US005781188A

United States Patent [19]

Amiot et al.

[11] Patent Number: 5,781,188

[45] Date of Patent: Jul. 14, 1998

[54] **INDICATING ACTIVENESS OF CLIPS AND APPLYING EFFECTS TO CLIPS AND TRACKS IN A TIMELINE OF A MULTIMEDIA WORK**

[75] Inventors: **Luc R. Amiot**, Boisbriand; **Jacques Y. Deveau**, Montreal; **Michael C. Sheasby**, London; **Peter H. Ibrahim**; **Raymond Hill**, both of Montreal, all of Canada; **Darryl M. Lewis**, Bellevue, Wash.

[73] Assignee: **Softimage**, Montreal, Canada

[21] Appl. No.: **673,223**

[22] Filed: **Jun. 27, 1996**

[51] **Int. Cl.**[6] **G06F 3/00**

[52] **U.S. Cl.** **345/328**; 345/302

[58] **Field of Search** 345/326–358, 345/302

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,148,154	9/1992	MacKay et al.	345/328
5,191,645	3/1993	Carlucci et al.	345/328
5,206,929	4/1993	Langford et al.	345/328
5,339,393	8/1994	Duffy et al.	345/328
5,359,712	10/1994	Cohen et al.	345/328
5,388,197	2/1995	Rayner	345/328
5,404,316	4/1995	Klingler et al.	345/328 X
5,442,744	8/1995	Piech et al.	345/328
5,467,288	11/1995	Fasciano et al.	345/328 X
5,537,528	7/1996	Takahashi et al.	345/328 X
5,644,216	7/1997	Blumenau	345/328 X
5,664,087	9/1997	Tani et al.	345/328 X
5,682,326	10/1997	Klingler et al.	345/328 X

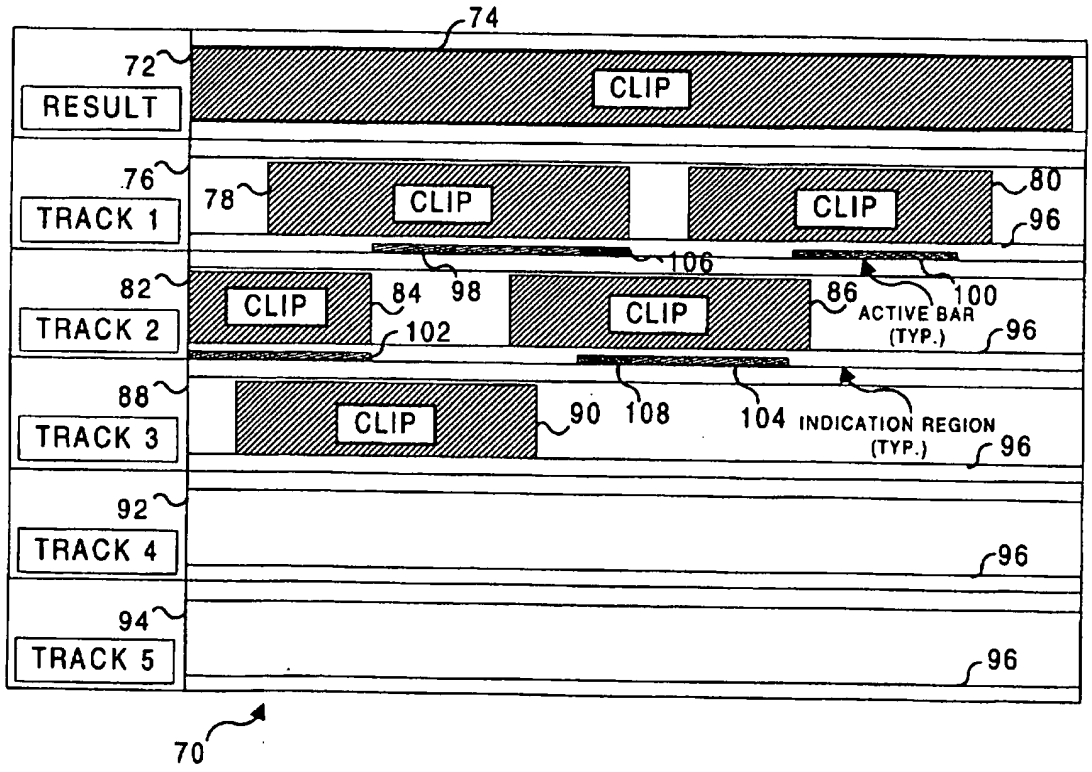
Primary Examiner—John E. Breene

Attorney, Agent, or Firm—Ronald M. Anderson

[57] **ABSTRACT**

For use in composing and/or editing a multimedia work that include one or more clips, a technique for indicating an active state of the clip or portion thereof, indicating effects applied to a track or a clip, and indicating a plurality of effects applied to a track or to a clip. At least a portion of the work is graphically represented by one or more timelines on which are disposed one or more clips. An active portion of a clip is indicated by a colored bar that extends under that portion of the clip. Different types of transitions between clips are indicated by using one of different predetermined colors for the colored bar. Effects applied to a track are also represented by color bars that extend above the portion of the track to which the effects apply and are independent of changes to any clip disposed on the track. Predetermined colors are used to indicate the type or class of effect(s) applied. A plurality of effects applied to a clip or to a track are indicated by stacking the colored bars. The duration of the effect(s) are indicated by the relative length of these colored bars.

46 Claims, 11 Drawing Sheets



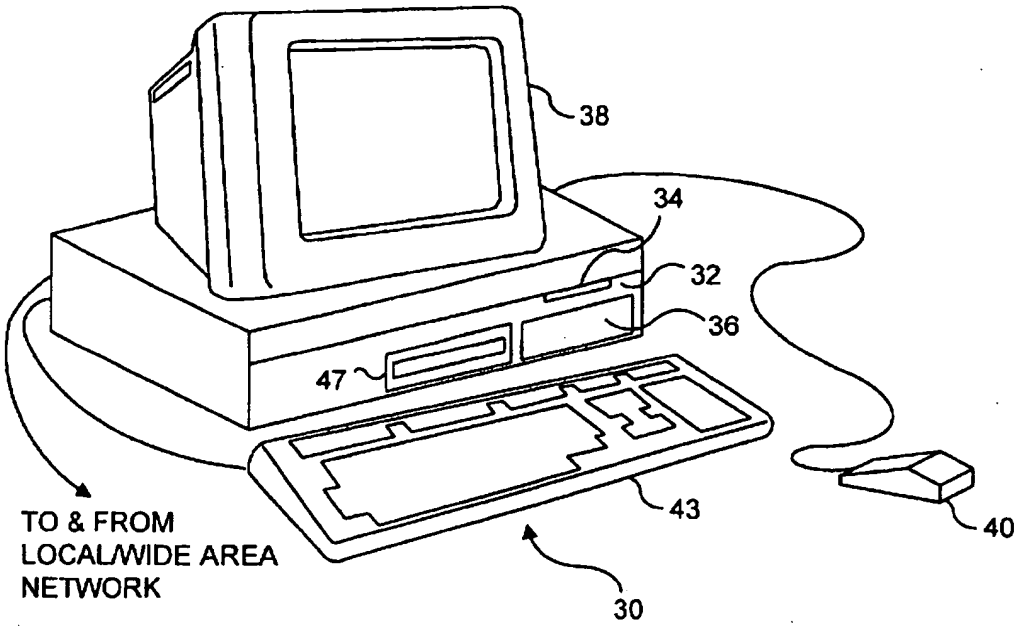

38
34
32
36
47
43
40
TO & FROM
LOCAL/WIDE AREA
NETWORK
30

FIG. 1

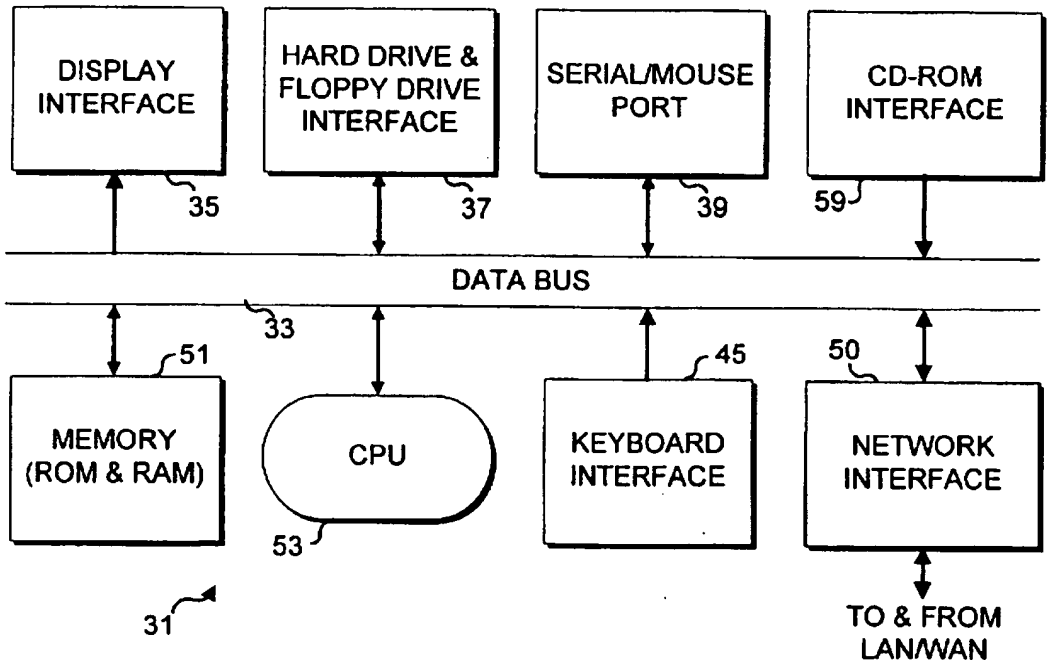

DISPLAY INTERFACE
HARD DRIVE & FLOPPY DRIVE INTERFACE
SERIAL/MOUSE PORT
CD-ROM INTERFACE
35
37
39
59
DATA BUS
33
51
45
50
MEMORY (ROM & RAM)
CPU
KEYBOARD INTERFACE
NETWORK INTERFACE
53
31
TO & FROM
LAN/WAN

FIG. 2

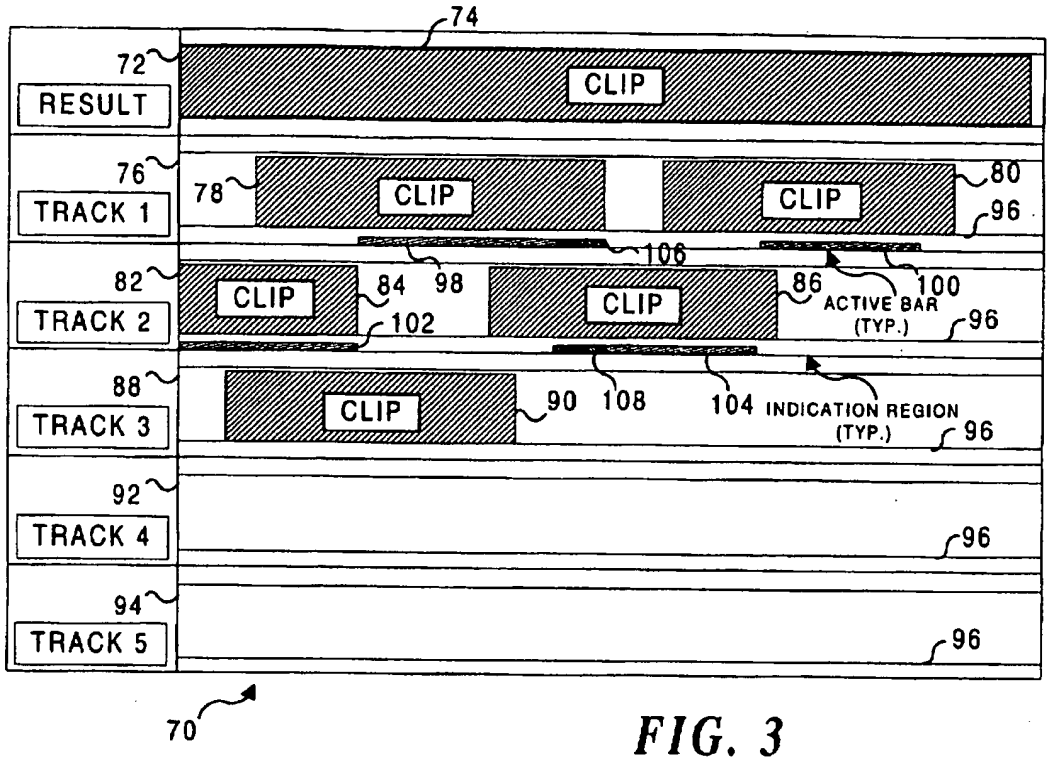


FIG. 3

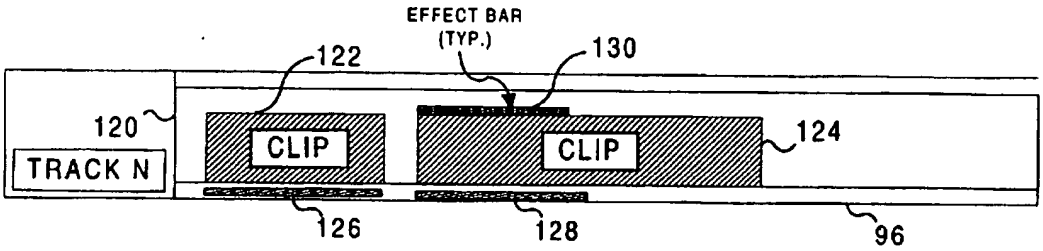


FIG. 4

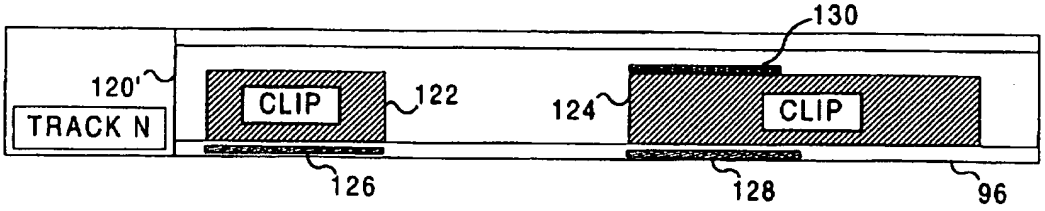


FIG. 5

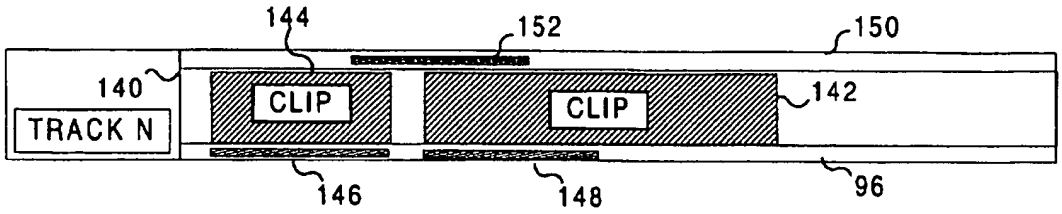


FIG. 6

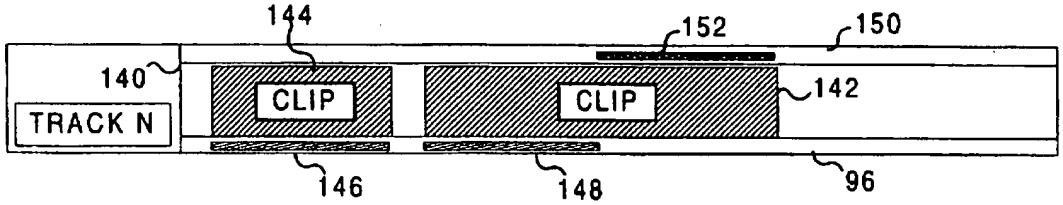


FIG. 7

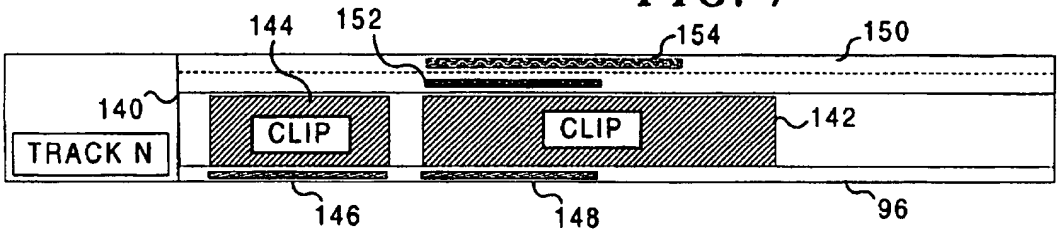


FIG. 8

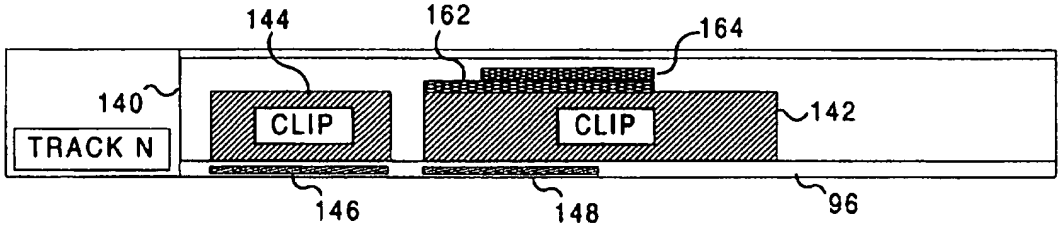


FIG. 9

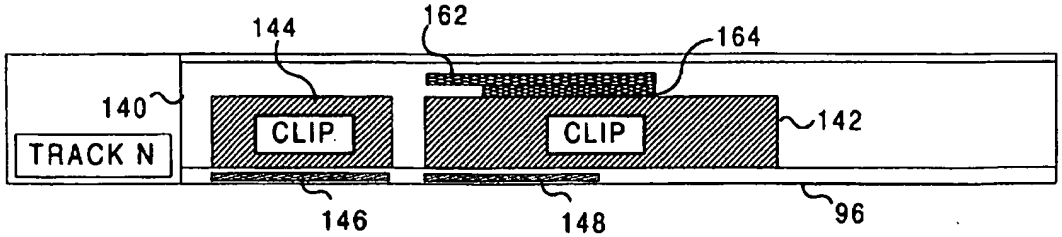


FIG. 10

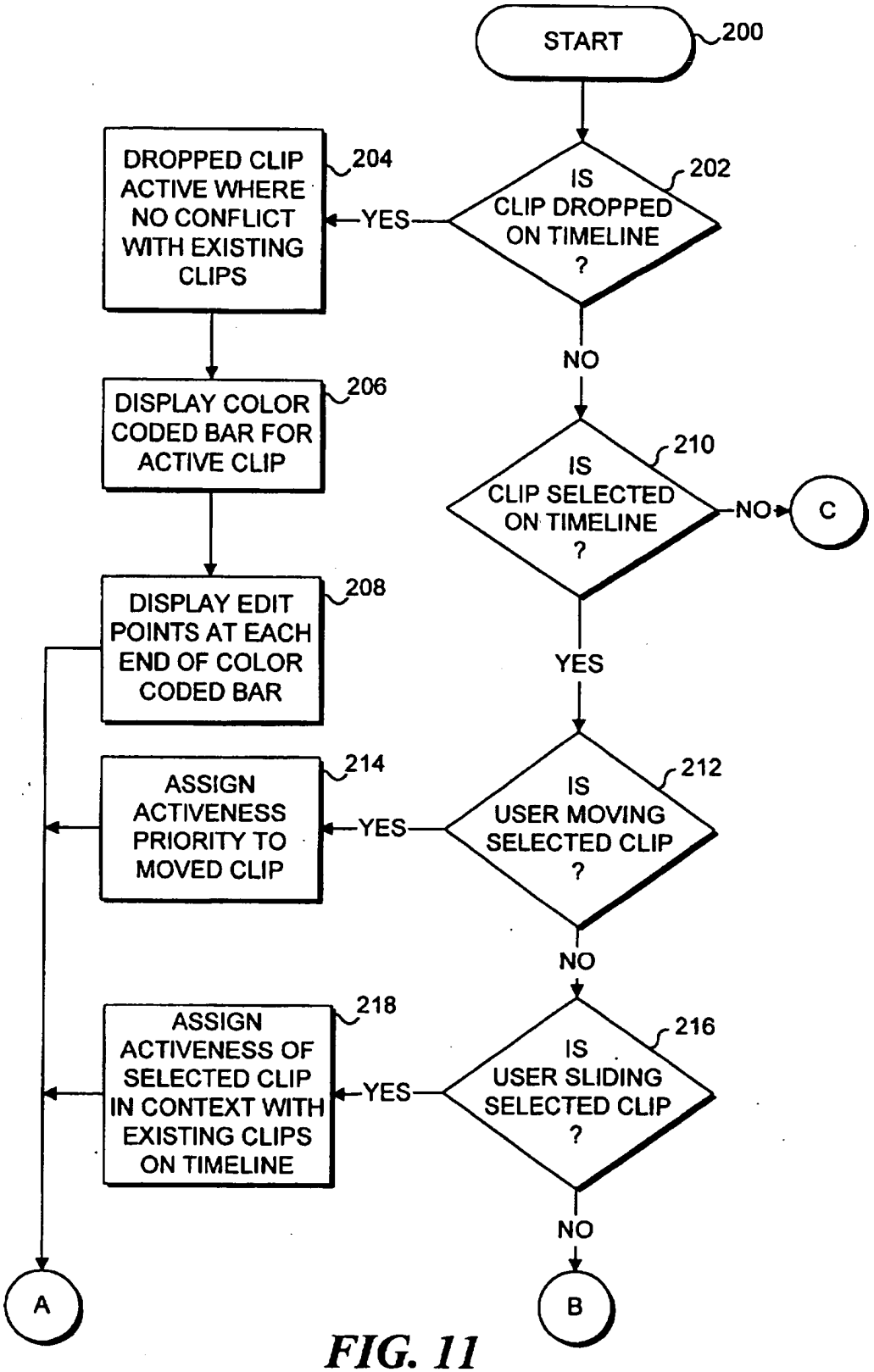

START
200
IS CLIP DROPPED ON TIMELINE ?
202
YES
DROPPED CLIP ACTIVE WHERE NO CONFLICT WITH EXISTING CLIPS
204
NO
DISPLAY COLOR CODED BAR FOR ACTIVE CLIP
206
IS CLIP SELECTED ON TIMELINE ?
210
NO
C
YES
DISPLAY EDIT POINTS AT EACH END OF COLOR CODED BAR
208
IS USER MOVING SELECTED CLIP ?
212
YES
ASSIGN ACTIVENESS PRIORITY TO MOVED CLIP
214
NO
IS USER SLIDING SELECTED CLIP ?
216
YES
ASSIGN ACTIVENESS OF SELECTED CLIP IN CONTEXT WITH EXISTING CLIPS ON TIMELINE
218
NO
A
B

FIG. 11

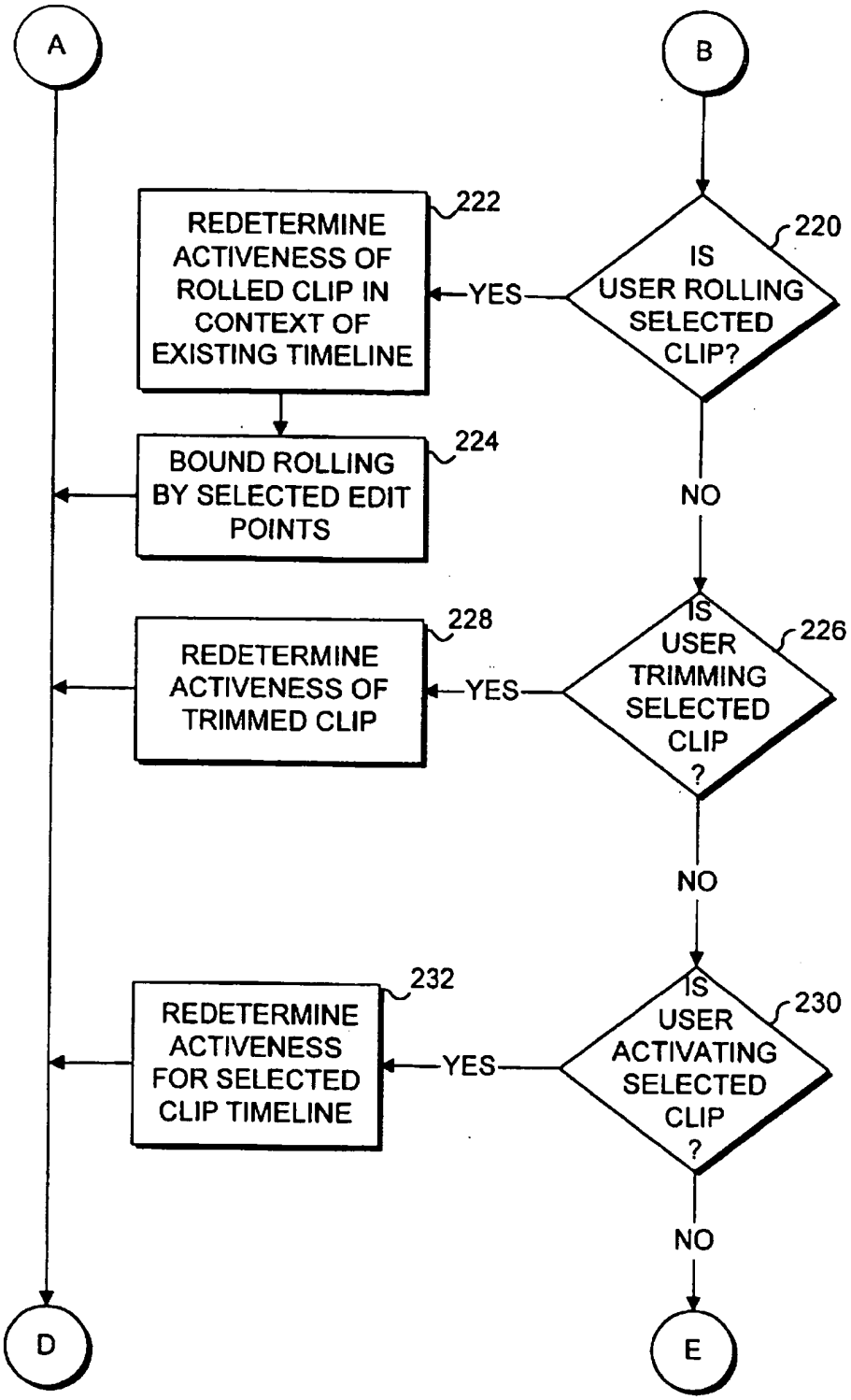

A
B
REDETERMINE ACTIVENESS OF ROLLED CLIP IN CONTEXT OF EXISTING TIMELINE
222
YES
IS USER ROLLING SELECTED CLIP?
220
BOUND ROLLING BY SELECTED EDIT POINTS
224
NO
REDETERMINE ACTIVENESS OF TRIMMED CLIP
228
YES
IS USER TRIMMING SELECTED CLIP ?
226
NO
REDETERMINE ACTIVENESS FOR SELECTED CLIP TIMELINE
232
YES
IS USER ACTIVATING SELECTED CLIP ?
230
NO
D
E

FIG. 12

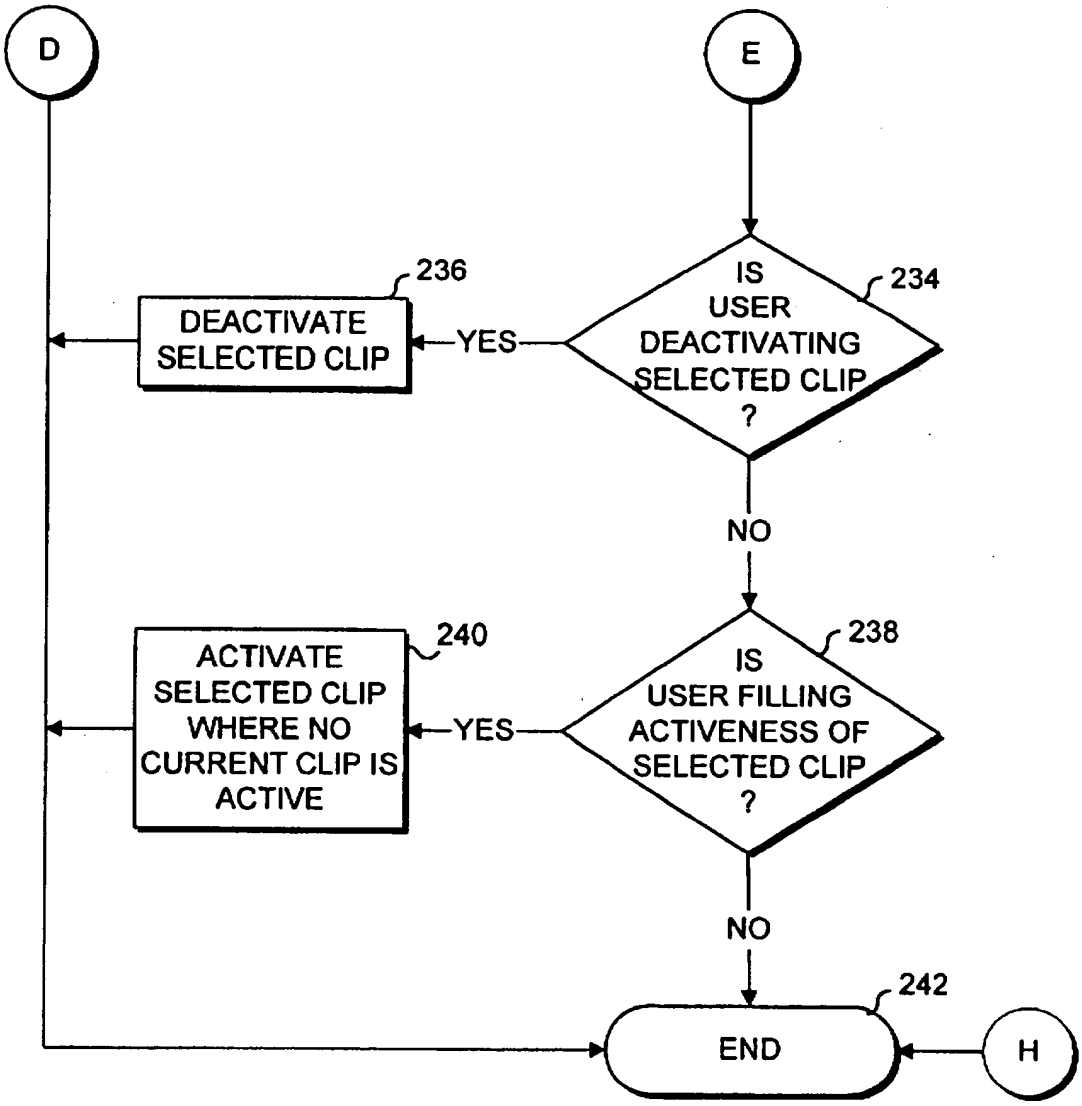

D
E
IS USER DEACTIVATING SELECTED CLIP ?
234
YES
DEACTIVATE SELECTED CLIP
236
NO
IS USER FILLING ACTIVENESS OF SELECTED CLIP ?
238
YES
ACTIVATE SELECTED CLIP WHERE NO CURRENT CLIP IS ACTIVE
240
NO
END
242
H

FIG. 13

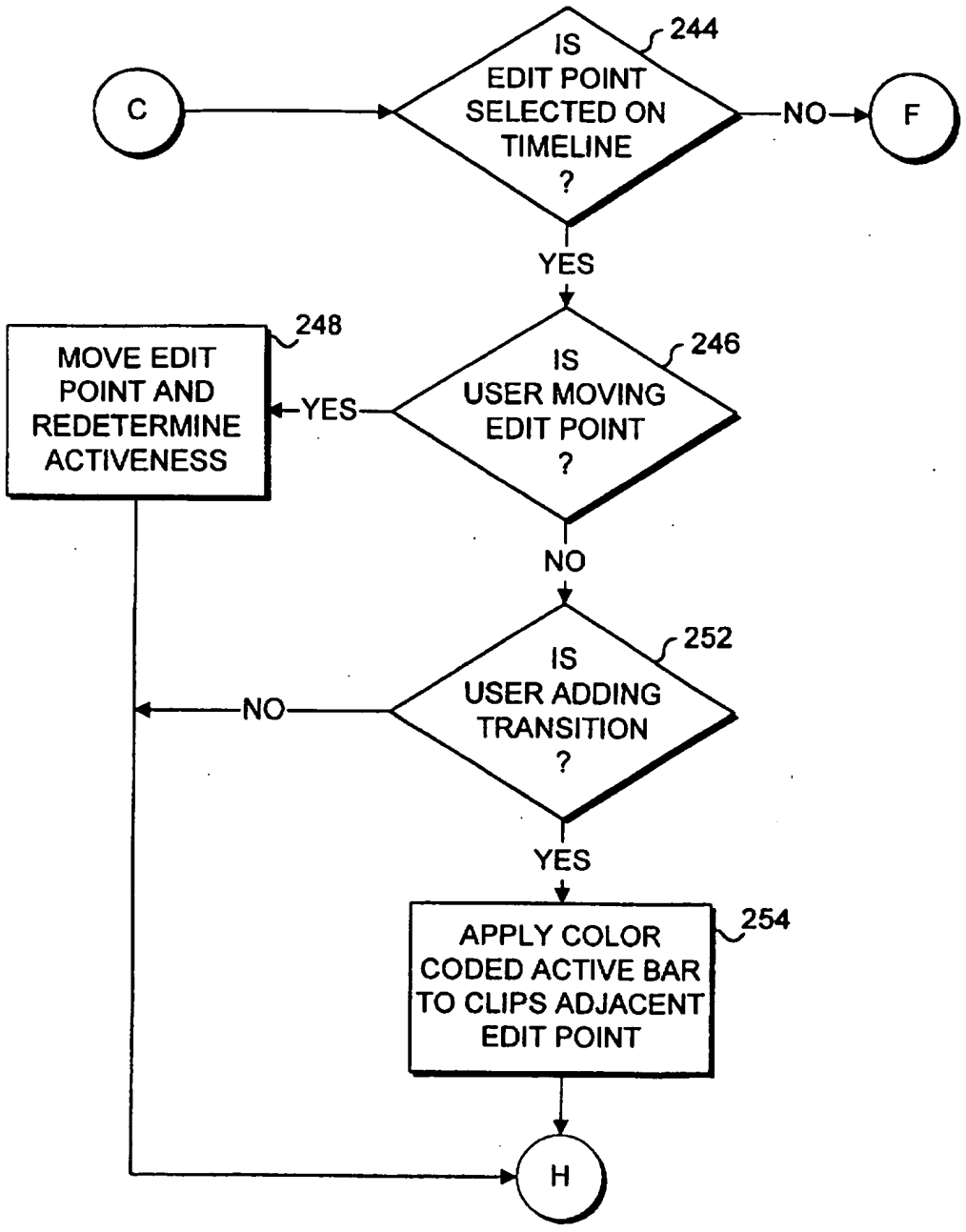

C
IS EDIT POINT SELECTED ON TIMELINE ?
244
NO
F
YES
IS USER MOVING EDIT POINT ?
246
YES
MOVE EDIT POINT AND REDETERMINE ACTIVENESS
248
NO
IS USER ADDING TRANSITION ?
252
NO
YES
APPLY COLOR CODED ACTIVE BAR TO CLIPS ADJACENT EDIT POINT
254
H

FIG. 14

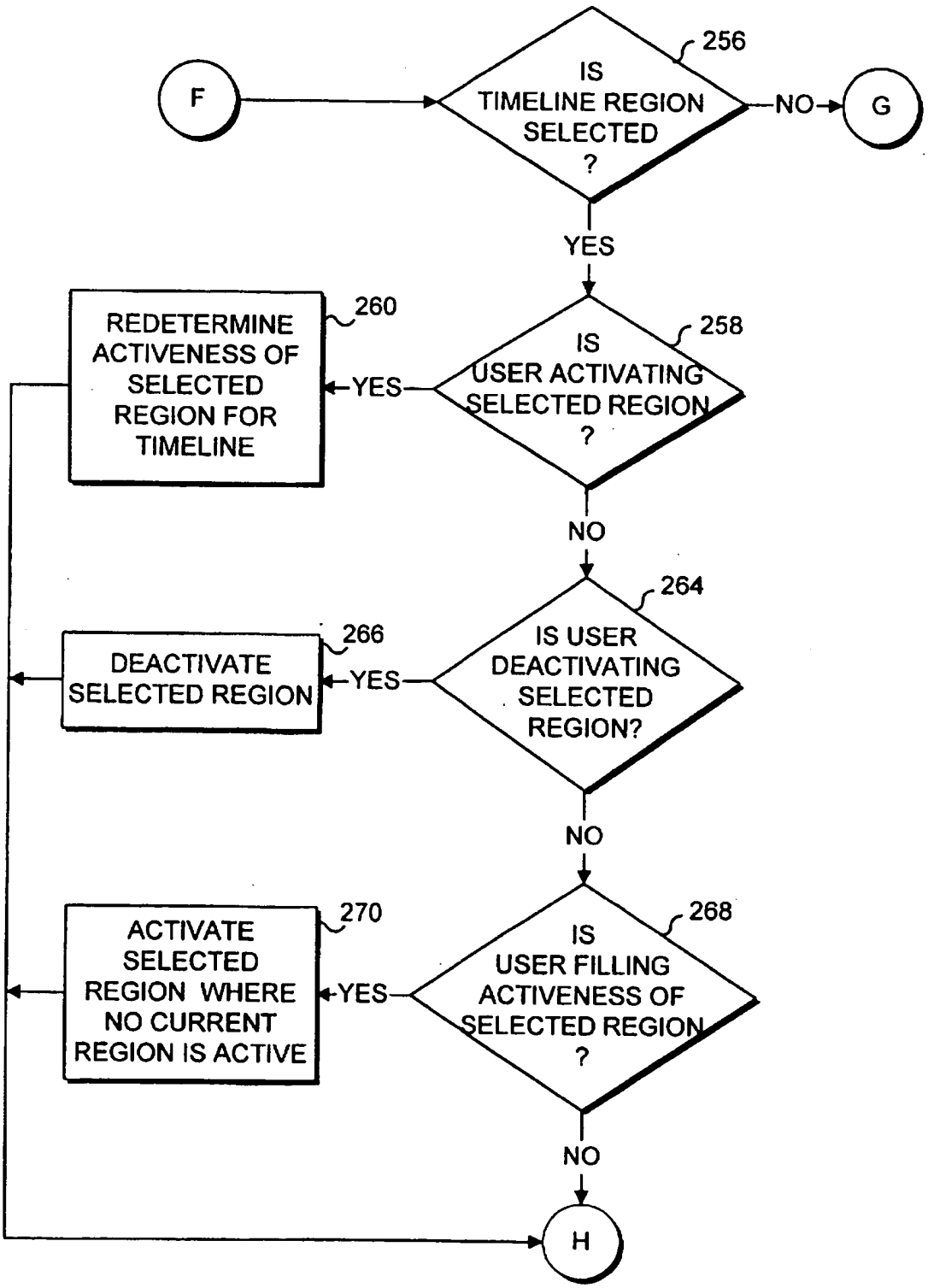

F
256
IS TIMELINE REGION SELECTED ?
NO
G
YES
260
REDETERMINE ACTIVENESS OF SELECTED REGION FOR TIMELINE
YES
258
IS USER ACTIVATING SELECTED REGION ?
NO
264
IS USER DEACTIVATING SELECTED REGION?
266
DEACTIVATE SELECTED REGION
YES
NO
270
ACTIVATE SELECTED REGION WHERE NO CURRENT REGION IS ACTIVE
YES
268
IS USER FILLING ACTIVENESS OF SELECTED REGION ?
NO
H

FIG. 15

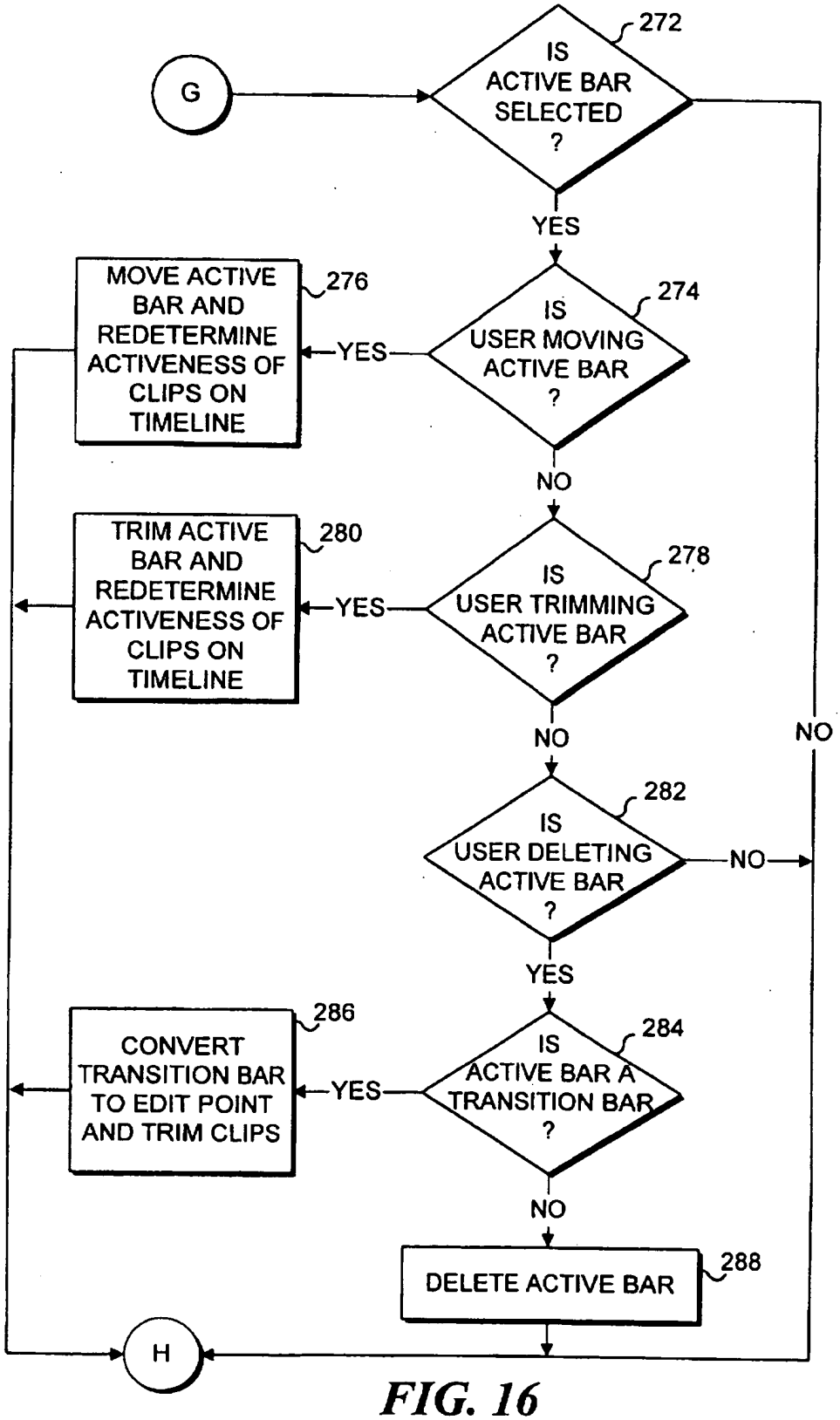

G
272
IS ACTIVE BAR SELECTED ?
YES
NO
274
IS USER MOVING ACTIVE BAR ?
YES
276
MOVE ACTIVE BAR AND REDETERMINE ACTIVENESS OF CLIPS ON TIMELINE
NO
278
IS USER TRIMMING ACTIVE BAR ?
YES
280
TRIM ACTIVE BAR AND REDETERMINE ACTIVENESS OF CLIPS ON TIMELINE
NO
282
IS USER DELETING ACTIVE BAR ?
NO
YES
284
IS ACTIVE BAR A TRANSITION BAR ?
YES
286
CONVERT TRANSITION BAR TO EDIT POINT AND TRIM CLIPS
NO
288
DELETE ACTIVE BAR
H

FIG. 16

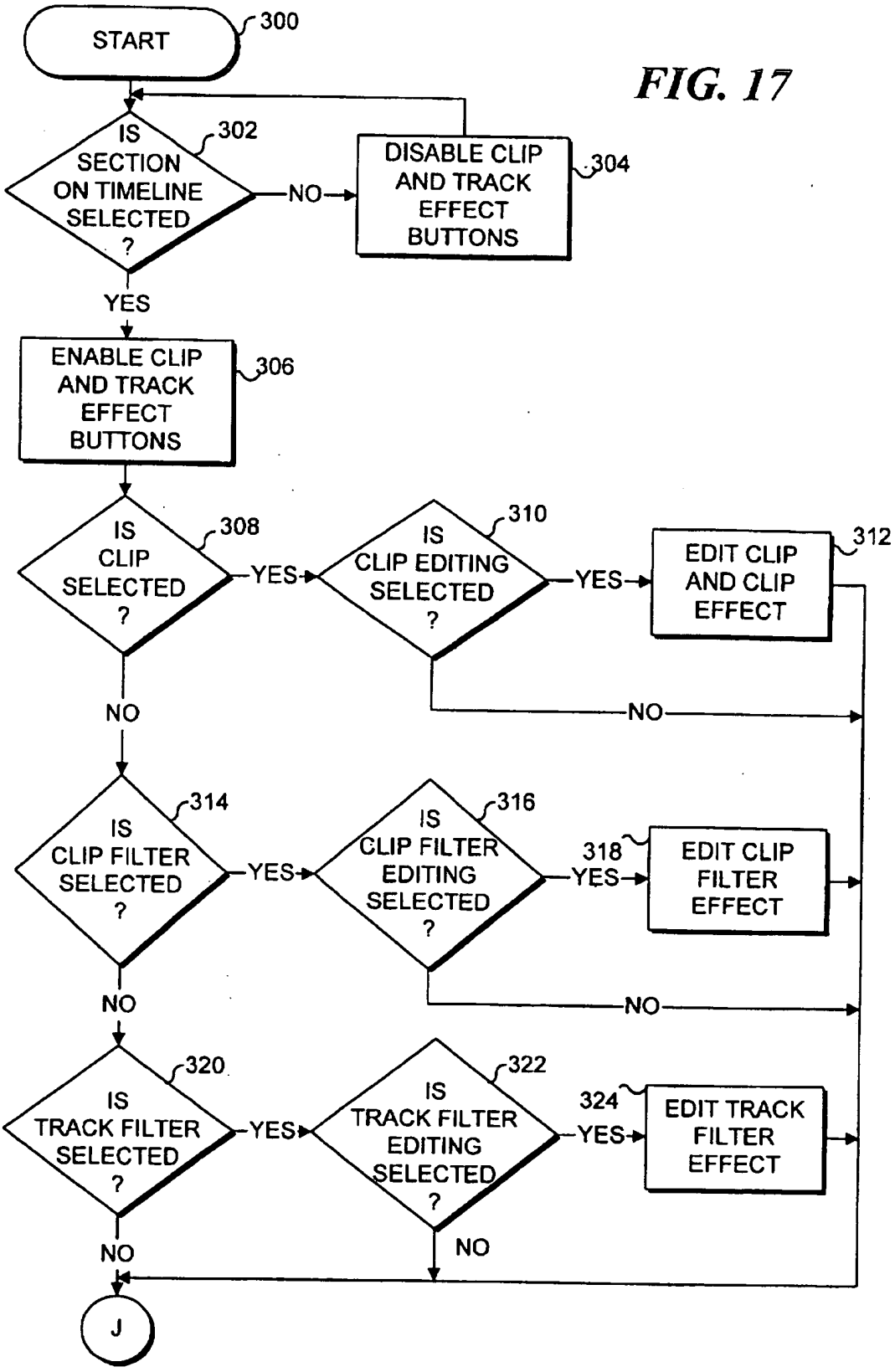

START
300
IS SECTION ON TIMELINE SELECTED ?
302
NO
DISABLE CLIP AND TRACK EFFECT BUTTONS
304
YES
ENABLE CLIP AND TRACK EFFECT BUTTONS
306
IS CLIP SELECTED ?
308
YES
IS CLIP EDITING SELECTED ?
310
YES
EDIT CLIP AND CLIP EFFECT
312
NO
NO
IS CLIP FILTER SELECTED ?
314
YES
IS CLIP FILTER EDITING SELECTED ?
316
YES
318
EDIT CLIP FILTER EFFECT
NO
NO
IS TRACK FILTER SELECTED ?
320
YES
IS TRACK FILTER EDITING SELECTED ?
322
YES
324
EDIT TRACK FILTER EFFECT
NO
NO
J

FIG. 17

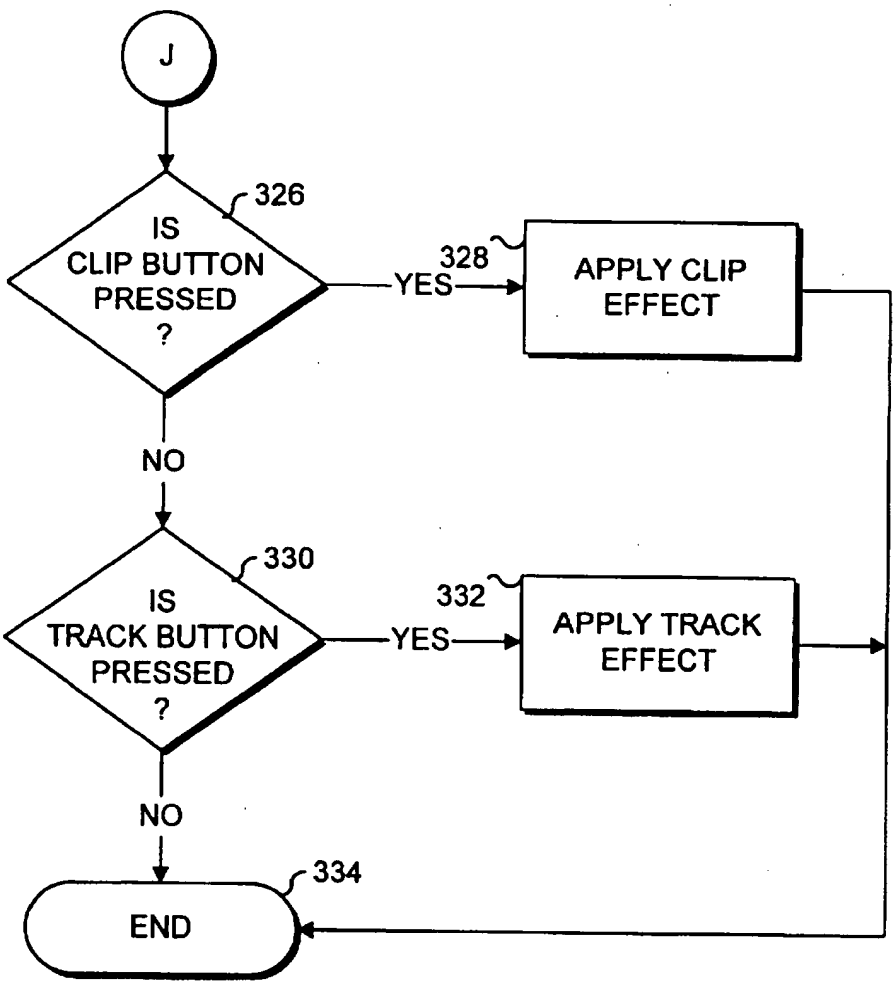

J
326
IS CLIP BUTTON PRESSED ?
YES
328
APPLY CLIP EFFECT
NO
330
IS TRACK BUTTON PRESSED ?
YES
332
APPLY TRACK EFFECT
NO
334
END

FIG. 18

INDICATING ACTIVENESS OF CLIPS AND APPLYING EFFECTS TO CLIPS AND TRACKS IN A TIMELINE OF A MULTIMEDIA WORK

FIELD OF THE INVENTION

This invention generally relates to a system for creating and modifying a multimedia work, and more specifically, to a system and method for indicating clips that are active and for indicating effects that are applied to components of the multimedia work.

BACKGROUND OF THE INVENTION

When creating a multimedia work that may include a plurality of different components such as animations, video clips, and audio clips, authoring software typically provides a multitude of options for modifying various aspects of the components comprising the work. The components of a graphics sequence created on a computer may comprise a plurality of clips, each of which typically comprises a plurality of frames. When displayed at a sufficiently fast rate, the objects shown in different positions or attitudes in successive frames appear to move, creating an animation. A similar approach is used in video clips, except that the frames of a video clip comprise digitized images rather than graphic illustrations such as those that are typically produced by a two or three-dimensional paint program.

A multimedia work is created by combining various clips. It is generally desirable to provide the user with an indication of the state of the clips loaded into the authoring software. Each of the prior art multimedia authoring software products uses a slightly different user interface to show a track and to indicate the state of the clips comprising a track. The temporal arrangement of clips is generally indicated by an arrangement of the clips along either a horizontal or a vertical axis and is referred to as a "timeline."

For example, in Quantel's HENRY™ software program, the timeline is vertical arrangement of clips. Successive clips that run sequentially appear in the vertical timeline with each successive frame being represented by a rectangle below the preceding frame in the clip. Likewise, each successive clip follows below the preceding clip. A color coded bar extends down the left side of each clip to indicate its state, e.g., the bar is yellow to indicate that a clip is active and that no other portion of any other clip is active. If two successive clips A and B overlap where a transition occurs between the clips, a different color bar is provided that extends along the side of only those frames of the clips that overlap in time.

In Avid's MEDIA COMPOSER™, the timeline is horizontal. Multiple clips can be shown on different tracks during overlapping times; however, vertical ordering of the clips determines the clip or part of a clip that is active at a given time. The program treats any portion of a clip on a track above a portion of another clip (located on a vertically lower track) as the active portion of the clip. Thus, the active portion of any clip is controlled by the relative vertical position of that clip compared to other clips. It will be apparent that the techniques used in the prior art to indicate the active clip or active portion of a clip in a non-linear editing system do not provide much versatility. Ideally, an operator should be able to see all options that are available for use in a work and be able to immediately perceive the options that are being used in the output signal.

An analogous problem arises during the live broadcast of a sports event in which multiple cameras disposed at different locations and capturing different elements of the event are all potentially a source of a network feed signal that is broadcast (or taped). The director of the production must select a particular camera for the signal output at any one time. All of the cameras feed to a central control panel for display on an array of monitors so that all of the views from which one can be selected for the output signal are visible. An indicator identifies which camera is currently active. This approach permits great flexibility and is extremely efficient in enabling an optimum output signal to be selected as the scene being televised or taped changes. Clearly, it would be desirable to extend this concept to multimedia authoring software to enable a user to view all of the clips from which the active clip or portion thereof can be selected and to indicate which portion of any clip is active. Currently, such features are not available.

Multimedia authoring systems typically provide a number of different effects that can be applied to components of the work to modify the output. The term "effect" refers to any processing applied to input data that in some manner alters the data. Effects such as dissolve, wipe, and various digital video effect transitions can be applied to control the appearance applied to a change from one clip to another. Other visual effects such as blur, color correction, and chroma key modify the overall visual appearance of a clip or portion thereof. Still other effects, such as echo, are applied exclusively to audio clips. Since a user can apply any of a rather large number of selected effects to change the result produced in most multimedia authoring software, it is important to provide a visual indication of the type or class of effect being applied and to clearly indicate the portion of a clip to which an effect will apply. The prior art interface schemes for indicating effects differs between each program. Color bars have been used to indicate that an effect is being applied to an entire track or to show the portion of a clip to which the effect applies, with the specific color indicating the type or class of effect being applied. However, there is no provision in the prior art for applying an effect to only a specific portion of a track that is less than the entire track so that the effect is independent of any clip that is placed in that portion. Enabling a user to associate an effect with a specific portion of a track rather than with a clip would have a significant advantage. The user would then be able to change the clip inserted on the track at that point, and the effect associated with that portion of the track would automatically apply to any clip that is thus inserted.

There are clearly times when it will be desirable to apply multiple effects to different portions of a track or a clip. Currently, multimedia authoring software does not provide an effective procedure for selectively applying multiple effects to limited portions of a track. Typically, in prior art multimedia authoring systems, an effect is applied to the entire duration of a clip. If the user wants to apply a first effect (1) to the first half of a clip and a second effect (2) only to the middle one-third of the clip, it will be necessary for the user to divide the clip into four new clips, A–D, where clip A is the first third of the original clip and has only effect (1) applied to it, clip B is the next ⅙ of the original clip and has both effects (1) and (2) applied to it, clip C is the next ⅙ of the original clip and has only effect (2) applied to it, and clip D is the last one third of the original clip and has no effects applied to it. It will be evident that dividing clips to facilitate applying effects to only a portion of a clip is inefficient and that a better approach is required to achieve the desired flexibility.

SUMMARY OF THE INVENTION

In accord with the present invention, a method for indicating activeness of a clip to a user composing and/or editing

a multimedia work that includes a plurality of clips provides for displaying a timeline for at least a portion of the multimedia work. The timeline graphically represents the portion of the multimedia work and graphically illustrates a plurality of tracks for a time interval corresponding to that represented by the timeline. The user is enabled to select a clip as active, indicating that the clip is to be used in the multimedia work. The clip selected by the user is marked as active with a visual indicator, so that by inspection of the visual indicator and a position of the clip on one of the plurality of tracks, the user can determine a time during which the clip contributes to the multimedia work.

The user is also enabled to select a portion of the clip as active. The portion of the clip selected as active is then marked with the visual indicator, so that the position of the visual indicator relative to the clip indicates the portion of the clip that is active. Each clip is preferably represented by a rectangle disposed on one of the timelines. A duration of the clip is indicated by either its length or its width. The visual indicator preferably comprises a colored bar disposed adjacent to one edge of the rectangle; the length of the colored bar and its position relative to the rectangle indicates the duration of the portion of the clip that is active. Any portion of a clip added to one of the timelines that does not overlap any portion of another clip that is already active is automatically made active.

Another step in the method provides for enabling the user to select two clips to be simultaneously active at one time when either a transition or an overlay is to occur between the two clips. A visual indicator of the duration of the transition or the overlay between the two clips is then provided. The visual indicator preferably comprises a color bar having a color that indicates a specific type of transition or the overlay between the clips.

There are several advantages to the technique used to indicate whether a clip is active in the present invention. Since both active and inactive clips can be placed on the timeline, the user can visually perceive the contents of the clips that are included to facilitate selecting a clip that will be active. The available options are clearly presented for the user to view, enabling selection of the clip or portion of a clip that will be used in the work.

Another aspect of the present invention is directed to a system for indicating activeness of a clip to a user composing and/or editing a multimedia work that includes a plurality of clips. The system includes a monitor for displaying the multimedia work and for displaying screens used to compose and edit the multimedia work. A memory is included for storing a plurality of machine instructions defining a software program employed for composing and editing the multimedia work. A processor, which is coupled to the memory and to the monitor, executes the machine instructions stored in the memory. The functions implemented by the system as defined by the machine instructions are generally consistent with the steps of the method described above.

Yet another aspect of the present invention relates to a method for indicating effects applied to a specific portion of a multimedia work when composing and/or editing the multimedia work. The method includes the step of enabling the user to select an effect that will apply to the multimedia work and to select a time interval during which the effect will be applied. The effect is associated with a selected track of the timeline by the user and is applied to only a selected portion of the track. An effect indicator indicates the portion of the track to which the effect applies and the time interval of the effect.

The track preferably includes an effect region that extends adjacent to the track. The effect indicator is disposed within the track region. In the preferred form of the invention, the effect indicator comprises a colored bar. A length of the colored bar and its position relative to the track indicates the time interval over which the effect is applied. The effect is applied to a portion of any clip positioned on the track with which the effect is associated, and which is adjacent the colored bar, but the effect is otherwise independent of any clip.

The method enables the user to modify the time interval over which the effect is applied. The modification may include changing a point in time at which the effect is applied and its duration.

In addition, the user can associate a plurality of effects with the track. Each effect can have a user selectable time interval that is independent of the time interval of other effects associated with the track. The plurality of effects are represented by corresponding color bars that are disposed adjacent to the track with which the plurality of effects are associated. A length and a position of each of the color bars indicates the time and duration at which each is applied to the track. Different colors are preferably used for the color bars, each color representing a different type or class of effect.

The order in which the plurality of effects are applied to the track determines the result in the multimedia work. The color bars representing the plurality of effects are thus positioned relative to each other so as to indicate the order in which the effects are applied.

Another aspect of the present invention is directed to a system for indicating effects applied to a specific portion of a multimedia work when composing and/or editing the multimedia work in which a processor executes machine instructions that cause the processor to implement effects generally as described above.

Yet another aspect of the present invention is directed to a method for indicating a plurality of effects that are applied to clips comprising a multimedia work, when composing and/or editing the multimedia work. The method includes the step of enabling the user to select a plurality of different effects that will apply to a clip of the multimedia work and to select a portion of the clip to which each effect will be applied. An effect indicator is provided for each effect to indicate the portion of the clip to which the effect applies and the time interval during which the effect applies.

The effect indicator for each clip preferably comprises a colored bar, a length of the bar and its relative position adjacent the clip indicates the portion of the clip to which the effect applies. The colored bars representing each of the plurality of effects selected by the user for application to the clip are stacked adjacent to the clip. An order in which the colored bars are stacked indicates a sequence in which the effects are applied to the clip to produce a result. The plurality of effects are associated with the clip to which the plurality of effects apply, so that moving the clip along the timeline causes movement of the effects applied to the clip. The method also preferably includes the step of displaying a descriptive label when a cursor is moved over any effect indicator, said descriptive label identifying a type or class of effect that is represented by the effect indicator.

Still another aspect of the invention is a system for indicating a plurality of effects that are applied to clips comprising a multimedia work, when composing and/or editing the multimedia work. A processor executes machines instructions that cause it to implement functions generally consistent with the steps of the corresponding method discussed above.

Yet a further aspect of the invention is directed to an article of manufacture that includes a nonvolatile memory device used to store a plurality of machine instructions. The machine instructions generally cause a computer to implement the functions described above.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIG. **1** is an isometric view of a personal computer suitable for implementing the present invention;

FIG. **2** is a schematic block diagram of functional components included within a processor chassis for the personal computer of FIG. **1**;

FIG. **3** illustrates a timeline window that includes a result track and up to five component tracks, in which activeness of a portion of a clip on a track is indicated in accord with the present invention;

FIG. **4** is a portion of a track from a timeline window showing clip activeness and an effect applied to a portion of a clip are indicated;

FIG. **5** is the track of FIG. **4**, showing one of the clips shifted to a later point along the timeline;

FIG. **6** is a track to which a track-based effect has been applied;

FIG. **7** illustrates the change in the timeline of FIG. **6** that occurs when the track-based effect is shifted to a later point in time;

FIG. **8** illustrates an timeline in which a track-based effects stack is used;

FIG. **9** illustrates a timeline in which an effects stack is applied to a clip;

FIG. **10** shows a track in which the order of the effects in the stack applied to the clip of FIG. **9** is changed;

FIGS. **11** through **16** are a flow chart showing the logical steps used for determining activeness of clips and portions of clips on timelines; and

FIGS. **17** and **18** are a flow chart showing the logic used in the present invention for handling stacked effects applied to clips and tracks.

DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference to FIG. **1**, a generally conventional personal computer **30** is illustrated, which is suitable for use in connection with practicing the present invention. Alternatively, a workstation may be instead be used. Personal computer **30** includes a processor chassis **32** in which are mounted a floppy disk drive **34**, a hard drive **36**, a motherboard populated with appropriate integrated circuits (not shown), and a power supply (also not shown), as are generally well known to those of ordinary skill in the art. A monitor **38** is included for displaying graphics and text generated by software programs that are run by the personal computer. A mouse **40** (or other pointing device) is connected to a serial port (or to a bus port) on the rear of processor chassis **32**, and signals from mouse **40** are conveyed to the motherboard to control a cursor on the display and to select text, menu options, and graphic components displayed on monitor **38** by software programs executing on the personal computer. In addition, a keyboard **43** is coupled to the motherboard for user entry of text and commands that affect the running of software programs executing on the personal computer.

Personal computer **30** also optionally includes a compact disk-read only memory (CD-ROM) drive **47** into which a CD-ROM disk may be inserted so that executable files and data on the disk can be read for transfer into the memory and/or into storage on hard drive **36** of personal computer **30**.

FIG. **2** is a block diagram illustrating some of the functional components that are included within processor chassis **32**. All of the components mounted on the motherboard or otherwise installed inside processor chassis **32** are not illustrated, but the functional blocks shown in FIG. **2** represent the more significant aspects of the personal computer. The motherboard includes a data bus **33** to which these functional components are electrically connected. A display interface **35**, for example, a video card, generates signals in response to instructions executed by a CPU **53** that are transmitted to monitor **38** so that graphics and text are displayed on the monitor. A hard drive and floppy drive interface **37** is coupled to data bus **33** to enable bidirectional flow of data and instructions between data bus **33** and floppy drive **34** or hard drive **36**. Software programs executed by CPU **53** are typically stored on either hard drive **36**, or on a floppy disk (not shown) that is inserted into floppy drive **34**. The software program comprising the present invention will likely be distributed either on such a floppy disk, on-line via the modem, or on a CD-ROM disk, and stored on hard drive **36** for execution by CPU **53**.

A serial/mouse port **39** (representative of the two serial ports typically provided) is also bidirectionally coupled to data bus **33**, enabling signals developed by mouse **40** to be conveyed through the data bus to CPU **53**. Alternatively, if a different device such as an external modem (not shown) is coupled to the serial port, data can be transmitted bidirectionally between the CPU and the modem. A CD-ROM interface **59** connects CD-ROM drive **47** to data bus **33**. The CD-ROM interface may be a small computer systems interface (SCSI) type interface or other interface appropriate for connection to and operation of CD-ROM drive **47**.

A keyboard interface **45** receives signals from keyboard **43**, coupling the signals to data bus **33** for transmission to CPU **53**. Also coupled to data bus **33** is a network interface **50** (which may comprise an Ethernet card for coupling the personal computer to a local area and/or wide area network). Data used in connection with the present invention may be stored on a server and transferred to personal computer **30** over the network to carryout the functions described below.

When a software program is selected by a user to be executed by CPU **53**, the machine instructions comprising the program that are stored on a memory device such as a floppy disk, a CD-ROM, or on hard drive **36** are transferred into a memory **51** via data bus **33**. Machine instructions comprising the software program are executed by CPU **53**, causing it to implement functions defined by the machine instructions. Memory **51** includes both a nonvolatile read only memory (ROM) in which machine instructions used for booting up personal computer **30** are stored, and a random access memory (RAM) in which machine instructions and data are temporarily stored when executing application programs. The present invention will likely be included in a multimedia authoring software program tentatively called DIGITAL STUDIO™ to be distributed by SoftImage Corporation, a subsidiary of Microsoft Corporation. Due to

the size and the number of files comprising this software program, it will preferably be distributed on CD-ROM, but may alternatively be distributed on floppy disks.

With reference to FIG. 3, a timeline window 70 that is displayed on monitor 38 when editing a multimedia work in accord with the present invention is shown. Timeline window 70 includes a plurality of vertically arrayed, parallel tracks 72, 76, 82, 88, 92, and 94. Track 72 includes a result or composite clip 74, which represent the output of the work for the portion displayed in timeline window 70; however, a result track is not always present. Time extends from left to right in the timeline window. Each clip comprises a plurality of frames (not separately shown). Track 76 includes two clips 78 and 80, which are not contiguous. Only a portion of each of these clips is active, i.e., used in result clip 74, as explained below. Track 82 includes two clips 84 and 86, but only a portion of clip 86 is used in the result clip. Track 88 includes a single clip 90, but clip 90 is not currently used in the result clip.

Along the bottom of each track extends a region 96 in which the activeness of any portion of a clip in a track is indicated by a color coded bar extending below the active portion of the clip. Thus, in track 76, a yellow bar 98 extends below most of the latter two-thirds of clip 78, indicating that portion under which the yellow bar extends is active and included in result clip 74. (Note that other color coding protocols can alternatively be employed besides those used in this preferred embodiment.) A yellow bar 100 also extends under a portion of clip 80, indicating the portion of clip 80 that is active and included in result clip 74. In track 82, a yellow bar 102 extends under the entire length (duration) of clip 84 in region 96, indicating that the entire content of clip 84 is used in result clip 74. It should be noted that since a timeline can theoretically be of infinite duration, a bar appearing more like a point may be appropriate to represent the duration of a short clip on the timeline, depending on the time resolution applied. A yellow bar 104 extends under most of clip 86, similarly indicating the portion of that clip, which appears in the result clip.

Part of the activeness bar under clip 78 overlaps with part of the activeness bar under clip 86. The portions of each of the two activeness bars that overlap correspond to the duration of a transition between the two overlapping clips. The transitions are represented by different colored (e.g., orange) bar 106 under the right end of clip 78 and by an identically colored (orange) bar 108 under a portion (toward the left end) of clip 86. Generally, except at transitions, only one clip or portion thereof is indicated as being active at one time on the timeline. (This rule applies to non-audio clips; however, multiple audio clips may be active at one time. The multiple audio clips are mixed together during the time of their overlap. A further distinction applies to synchronized audio clips. Any audio clip that is synchronized to a video clip is active only during the time that its associated video clip is active.)

During a visual transition such as a cross-fade or dissolve, two clips are concurrently active. Specific colors are preferably used to indicate each different type of transition, e.g., orange for a dissolve, green for a wipe, and blue for a digital video effect. Even though a clip is not active, such as clip 90, it can still appear on the timeline, enabling a user composing or editing the work to more readily decide whether any portion of the clips on the timeline should be used in the result clip. By enabling the user to visually see all of the clips on the timeline, including those that are not active and thus not included in the result, the user can much more efficiently decide which clips or portions of clips to include and how to configure cuts and transitions between clips. (Optionally, a user may turn-off the visibility of inactive clips and other material so that the only the material contributing to the output is visible.)

When a new clip is "dropped" onto the timeline using a conventional "drag and drop" technique or entered by selecting an option in the menu of the multimedia authoring and editing software, the clip automatically becomes active if no other clip already on the timeline is currently active in the time interval within which the clip is dropped. If dropped at a point on the timeline so that it overlaps a portion of a currently active clip, only the part of the dropped clip that does not overlap the already active portion of another clip becomes active. Those of ordinary skill in the art will appreciate that different rules can readily be applied to handle the drag and drop of a clip on the timeline. For example, a clip that is dropped may be made active over its entire duration, altering the activeness of clips previously active during that time.

The preceding comments are not applicable to audio. In a preferred form of the present invention, there are eight channels of audio, where each mono stream corresponds to a channel; a stereo track has two channels that can be active in parallel and are thus not mutually exclusive, as is the case with video. The number of channels is limited to eight, so that all audio tracks can be active all the time.

In FIG. 3, dropping clip 84 in the position shown on the timeline will automatically cause the entire clip to become active, as indicated by yellow bar 102, since clip 84 does not overlap any portion of another clip that is already active. However, when clip 90 was then dropped onto the timeline, none of it automatically became active, since its entire duration overlaps other active portions of clips previously on the timeline.

The user can readily change the activeness of any clip or portion thereof using either options presented on a screen menu by the software program or more simply by selecting the activeness bar with a screen cursor (using mouse 40 or other pointing device) and dragging the activeness bar to extend or shorten its duration. The activeness bars cannot be dragged or otherwise extended beyond an end of a clip under which they lie, but they can be deleted from a clip or added to an existing clip if no overlap with another active clip is caused by that action. Transitions are created by selecting an edit point (the cut) on a clip and changing it to a transition, e.g., by selecting a "transition button" on a toolbar. This action results in one or more transition dialog boxes opening on the monitor, in which the user can select the type of transition to be applied and other parameters relating thereto. Although not shown, it is possible to have transitions between clips disposed on the same track.

Turning now to FIG. 4, a track 120 is illustrated in which two clips 122 and 124 are included. A yellow bar 126 extends under the entire length of clip 122, and a yellow bar 128 extends under approximately the left half of clip 124, indicating that all of clip 122 is active, and about the first half of clip 124 is active. In addition, an effects bar 130 extends over the top left half of clip 124. Effects bar 130 is preferably color coded to represent a particular effect applied to this portion of clip 124. The effect can be any of a number of different types of effects as are commonly applied in digital editing programs, such as blur, convolution, color correction, etc.

Effect 130 is clip-based and is associated with a single clip. In FIG. 5, a track 120' is shown, which is identical to track 120, except that in track 120', clip 124 has been shifted

to a later point in time (along with its activeness bar). When clip **124** is moved along the timeline in this manner, effect bar **130** is automatically moved with the clip. The length of effect bar **130** determines the portion of clip **124** to which the effect applies. Using the screen cursor or specified keyboard shortcuts, a user can lengthen or shorten the effect duration and can shift the effect to apply to any portion of clip **124**. However, the effect cannot extend beyond the beginning or end of a clip with which it is associated and is automatically removed if the clip is removed from the track.

In contrast to the preceding example illustrating the use of clip-based effects, FIGS. **6** and **7** show a track-based effect applied to a track **140**. Track **140** includes clips **144** and **142**. As indicated by a yellow bar **146** extending under clip **144** and a yellow bar **148** extending under the first half of clip **142**, all of clip **144** is active, and the first half of clip **148** is active. Along the top of track **140** is a track effects region **150**. Within the track effects region is an effect bar **152** that extends over parts of clips **142** and **144**. Again, effect bar **152** can represent virtually any of a multitude of different effects commonly applied to portions of a multimedia work. If clips **142** and **144** are audio clips, the effect bar will represent an effect appropriately applied to an audio clip, such as echo, compression, or equalization. If the clips are video or three dimensional paint clips, the effect will be visual in nature, as appropriate for that type of clip. The duration of the effect represented by effect bar **152** is indicated by the relative length of the effect bar. The type or class of effect is indicated by a predetermined color coding scheme, generally as noted above in connection with clip-based effects. It is also contemplated that other types of visual effect indicators, such as graphic icons, could be used to indicate each type or class of effect. Further, such other types of visual indicators can be grouped other than in vertical stacks. For example, if a timeline is vertical, the group of icons or other visual indicators could be arrayed horizontally adjacent to the timeline.

Effect bar **152** is associated with track **140** and is thus independent of the clips placed on the track or of their location or duration. As shown in FIG. **7**, the track-based effect bar can be moved along the timeline independent of any clip on the track. If clip **142** is removed, effect bar **152** is not changed. Any portion of a clip placed on track **140** under effect bar **152** will have the effect represented by the effect bar applied to that portion of the clip. Because effect bar **152** is track-based, a user can select the effect represented thereby, set up all of the parameters of the effect, and then change the duration and position of the effect bar in track effect region **150** independent of the clips that are loaded on the track. These clips can then readily be changed or modified, enabling the user to observe the effect applied to each clip or portion of a clip that is moved under effect bar **152**.

In FIG. **8**, track effect region **150** is modified to show stacked track-based effects. In this Figure, an effect bar **154** has been added above effect bar **152** and extends beyond the right end of effect bar **152**. The stacking order of effect bars **152** and **154** controls the order in which they are applied to modify any clips or portions thereof underlying both effect bars. Specifically, the effect represented by effect bar **152** will be applied first, e.g., to the first part of clip **142**, and then the second effect represent by effect bar **154** will be applied. Since effect bar **154** extends beyond effect bar **152**, only the effect it represents will apply to the portion of any clip underlying the portion of effect bar **154** that extends beyond the end of effect bar **152**. This arrangement provides considerable versatile control over the portions of any clip to which an effect applies.

In FIGS. **9** and **10**, clip-based effect stacks are illustrated comprising an effect bar **162** and an effect bar **164**. Each of these effect bars represents a different effect that is associated with clip **142**. The duration of each of the clip-based effects represented by effect bars **162** and **164** is indicated by the relative length of the effect bars. The stacking order determines the order in which the effects are applied to the clip with which they are associated, with the effect bar closest to the clip being applied first. Thus, in FIG. **9**, the effect represented by effect bar **162** is applied first to the underlying portion of clip **142**. Since effect bar **164** is shorter, the effect it represents is applied only to the right portion of the first half of clip **142** as changed by the effect represented by effect bar **162**. In FIG. **10**, the order of effect bars **162** and **164** and the processing to achieve the effects they each represent is reversed. The result achieved from this arrangement of stacked effects can differ significantly from the result obtained for the clip-based effect stack shown in FIG. **9**.

In FIGS. **11** through **16**, the logic flow for implementing selection and activeness in the present invention is illustrated. Referring now to FIG. **11**, in a start block **200**, the program begins by monitoring input devices for multimedia authoring commands provided by a user. Once a command or action has been received, the logic flow advances to a decision block **202**, where the program determines if user has dropped a clip on a timeline. If the action is "a dropped clip," then the logic flow advances to a logic block **204** and the program makes the clip active. A clip "dropped" onto the timeline using a conventional "drag and drop" technique (or entered by selecting an option in the menu of the program) automatically becomes active if no clip (or portion thereof) already on the timeline is currently active within the time interval in which the clip is dropped. If dropped at a point on a timeline that overlaps a portion of a currently active clip, only the part of the dropped clip that does not overlap the already active portion of another clip becomes active. In a logic block **206**, the program displays a color coded bar for that portion of the dropped clip that is now active.

Next, the logic flow advances to a logic block **208** and displays edit points at each end of the color coded bar applied to the dropped clip. After displaying the edit points, the logic flow advances to an end block **242**, where the program flow loops back to start block **200** to await another user action. However, if the program concludes in decision block **202** that a clip was not dropped on a timeline, the logic flow advances to a decision block **210**, where the program determines whether or not a clip has been selected on the timeline. If so, the logic flow advances to a decision block **212**, to determine whether the user is moving the selected clip on the timeline. If so, the logic advances to a block **214**, where the program assigns the moved clip priority over existing active clips and globally redetermines the activeness for the timeline. After reevaluating the clip activeness, the logic flow advances to end block **242**. At this point, the program normally loops back to start block **200** and awaits further user actions.

In decision block **212**, the program can also conclude that a selected clip was not moved, and the logic flow advances to a decision block **216**, where the program will ascertain whether the user has slid the selected clip along the timeline. If so, the logic advances to a block **218**, to redetermine the activeness of the sliding clip in context with existing clips and globally reevaluate the activeness of the timeline. Once the evaluation is complete, the logic advances to end block **242**, and then loops back to start block **200** to await yet another user action. However, the program can also con-

clude that a selected clip was not slid on the timeline in decision block **216**; in response, the logic advances to a decision block **220** as shown in FIG. **12**.

Referring now to FIG. **12** and decision block **220**, the program determines whether the user is rolling the activeness of the selected clip left or right within the constraints of the clip's existing activeness edit points. If so, the logic advances to a block **222**, where the program redetermines the activeness of the selected clip by giving it priority over existing clips on the timeline. Next, the logic advances to a block **224**, where the program globally redetermines the activeness of the timeline and binds the selected clip's activeness relative to the edit points. After completing these determinations, the logic advances to end block **242**, looping back to start block **200** to await another user action.

In decision block **220**, the program may conclude that the clip was not rolled. If not, the logic advances to a decision block **226**, where the program determines whether the user has selected the clip for trimming. If so, the logic advances to a block **228**, and the program trims the selected clip. After trimming, the activeness of the timeline is globally redetermined using the selected clip and all existing clips on the timeline. Once this evaluation is performed, the logic advances to end block **242** and loops back to start block **200** to await further user actions.

Each time that the activeness of the timeline is redetermined, the result depends upon the nature of the change effected by the user (whether trimming, inserting, or deleting a clip) and the status of one or more related parameters that have been selected by the user. For example, if a clip is dropped onto the timeline over an existing clip, the redetermination of activeness will either move the existing clip to make room for the new clip if a "ripple mode" parameter is selected, or will delete the existing clip if not. However, since the rules applied for redetermining the activeness of the timeline are outside the scope of the present invention, there seems little reason to discuss such details in this disclosure.

If the program concludes that a user is not trimming a selected clip in decision block **226**, the logic advances to a decision block **230**, where the program determines whether the user has activated the selected clip. If so, the logic advances to a block **232**, where the program activates the selected clip and globally redetenmines the activeness of the clip and the timeline. After the evaluation of activeness has been completed, the logic advances to end block **242** and back to the start of the procedure.

In decision block **230**, the program may conclude that the user was not activating the selected clip, and if so, advances to a decision block **234**. Referring now to FIG. **13** and decision block **234**, the program determines whether the user has deactivated the selected clip. If so, the logic advances to a block **236**, where the program deactivates the selected clip and globally redetermines the activeness of the timeline with regard to any remaining active clips. Next, the logic advances to end block **242** and loops back to start block **200** to await another action by the user.

However, if in decision block **234** the program determines that a user is not deactivating a selected clip, the logic advances to a decision block **238**, to determine whether a user is "filling the selected clip with activeness." If so, the logic advances to a block **240** and the program makes active (fills in) those parts of the clip where there is no current activeness along the timeline from other existing clips. Once the clip activeness is filled in, the logic advances to end block **242**, where the program flow loops back to start block **200** and awaits a further action by a user. However, if the program concluded in decision block **238** that a user has not selected a clip in which the activeness if to be filled, the logic advances to end block **242**, where the program loops back to start block **200** to await another action by the user.

Referring back to FIG. **12** and decision block **210**, if the program concludes that the user had not selected a clip on the timeline, then the logic advances to a decision block **244**. Referring now to FIG. **14** and decision block **244**, the program determines if an edit point has been selected on the timeline. If so, the logic advances to a decision block **246**, where the program determines if the user has moved the selected edit point. Once the program concludes an edit point was selected for moving, the logic advances to a block **248**, which provides for moving the selected edit point to the new location and then globally redetermines the activeness for the timeline. Next, the logic advances to end block **242**, with the subsequent loop back to start block **200**.

In decision block **246**, the program may determine that the user is not moving the selected edit point, and if so, the logic advances to a decision block **252**. In this decision block, the logic determines whether the user has added a transition to the selected edit point. If no transition is added, the logic advances to end block **242** and loops back to start block **200**. Alternatively, in decision block **252**, the program may decide that the user has added an edit point transition, and if so, the logic advances to a function block **254**, where the program adds color coded activeness bars representing the transition to the clips immediately adjacent to the selected edit point. Once the activeness bars have been applied, the logic advances to end block **242**, looping back to start block **200** to await a further action by the user.

Referring to decision block **252**, if an edit point was not selected on the timeline, the logic advances to a decision block **256**. Referring now to decision block **256** in FIG. **15**, the program determines whether a timeline region was selected by the user. If so, the logic advances to a decision block **258**, to determine if the user is activating the selected region. Once the program concludes the region has been activated, the logic advances to a block **260**. In block **260**, the program globally redetermines the selected region's timeline. After performing this evaluation, the logic advances to end block **242**.

At decision block **258**, the program may determine that the user has not activated the selected region and will then advance to a decision block **264**. In decision block **264**, the program determines whether the user has deactivated the selected region. If so, the logic advances to a block **266**, to deactivate the selected region. After deactivation, the logic advances to end block **242**. However, a negative response to decision block **264** indicates that the user did not deactivate the selected region. In this case, the logic advances to a decision block **268**, where the program determines whether the user has selected filling in the activeness of the region. If so, the logic advances to a block **270**, in which the region is made active in those areas where there is no conflict with current active regions on the timeline.

After filling in the selected region, the logic advances to end block **242**. Alternatively, if the program concludes in decision block **268** that the user is not filling the selected region with activeness, then the logic advances to end block **242**, where the program flow loops back to start block **200** to await further user actions.

Referring to decision block **256**, if the program decides that a region has not been selected, the logic advances to a decision block **272**. In this decision block, the program

determines if an active bar has been selected. Referring to FIG. 16 and decision block 272, if the program concludes that an active bar has not been selected, then the logic advances to end block 242. Alternatively, if the program concludes that an active bar has been selected, the logic advances to a decision block 274, where the program determines if the user has moved the selected active bar. If so, the logic advances to a block 276. In block 276, program moves the active bar to the new location selected by the user and globally redetermines the activeness of the timeline. After the activeness is redetermined, the logic advances to end block 242, looping back to start block 200 to await still further user actions.

A negative response to decision block 274 indicates that the user is not moving the selected active bar. In response, the logic advances to a decision block 278, where the program determines if the user is trimming the bar. If so, the logic advances to a block 280 to trim the selected active bar and then globally redetermines the activeness of the timeline. Afterwards, the logic advances to end block 242 and back to start block 200.

In decision block 278, if the user is not trimming the selected active bar, the logic advances to a decision block 282, where the program determines whether the user wants to delete the selected active bar. If the bar is not selected for deletion, the logic advances to end block 242. Alternatively, if the program concludes that the user does want to delete the selected active bar, the logic advances to a decision block 284. This decision block determines whether the selected active bar is a transition bar, and if so, the logic advances to a block 286. In block 286, the program converts the transition bar into an edit point by trimming the clips. After creating the edit point, the logic advances to end block 242. However, if the program concludes in decision block 284 that the selected active bar is not a transition bar, the logic advances to a block 288. In block 288, the program deletes the selected active bar. Once the bar is deleted, the logic advances to end block 242, looping back to start block 200 to await yet another action by the user.

Turning now to FIGS. 17 and 18, a clip and track based effects flow chart is illustrated. In FIG. 17, the logic begins with a start block 300 and advances to a decision block 302, to determines whether a section on the timeline has been selected, i.e., has a clip, a time span, or any other object on the timeline been selected that defines a region or time duration on the timeline. If no section on the timeline has been selected, the logic advances to a block 304, in which the program disables the clip and track effect buttons (or any other applicable user interface mechanism for selecting effects). After disabling the buttons, the logic loops back to decision block 302 to await the selection of something on a timeline by the user. Once a selection has been made on the timeline, the logic advances to a block 306, where the program enables the clip and the track effect buttons. After the buttons are enabled, the logic advances to a decision block 308. In this decision block, the program determines if a clip has been selected. If the response is affirmative, the logic advances to a decision block 310, in which the program decides whether the clip editing feature has been selected. If it has, the logic advances to a block 312, to edit the clip and clip effect. After editing, the logic advances to a decision block 326 as shown in FIG. 18. Further, if the program concluded in decision block 310 that a clip was not selected for editing, then the logic advances to decision block 326.

In the response to decision block 308 is negative, the logic advances to a decision block 314, where the program determines whether the user has selected a clip filter. If so, the logic advances to a decision block 316. In decision block 316, the program determines whether clip filter editing has been selected. If this type of editing is selected, the logic advances to a block 318 to edit the clip filter effect. Once the clip effect is edited, the logic advances to decision block 326 in FIG. 18. Further, if the program concludes in decision block 316 that a clip filter was not selected for editing, the logic also advances to decision block 326.

In decision block 314, the program may conclude that a clip filter was not selected and then advances to a decision block 320. At this point, the program determines whether a track filter was selected by the user. If the filter was selected, the logic advances to a decision block 322, to determine whether the track filter was selected by the user for editing. If so, the logic advances to a function block 324, and the program edits the track filter effect accordingly. Once the track filter effect is edited, the logic advances to decision block 326 in FIG. 18. Further, if the program concludes a track filter was not selected in decision block 320, the logic also advances to decision block 326.

At decision block 326 in FIG. 18, the program determines if a clip button has been pressed by a user. If so, the logic advances to a block 328, where the program applies the selected clip effect. After the effect has been applied, the logic advances to an end block 334 and loops back to start block 300 to await another action by the user. In decision block 326, the program may conclude that a clip button was not pressed and, in response, the logic advances to a decision block 330. In decision block 330, the program determines whether a track button has been pressed by the user. If so, the logic advances to a block 332, where the program applies the selected track effect. Once the track effect has been applied, the logic advances to end block 334, where the program loops back to start block 300 to await yet another action by the user. Further, the same result occurs if the program concludes in decision block 330 that a track button was not pressed.

Although the present invention has been described in connection with the preferred form of practicing it, those of ordinary skill in the art will understand that many modifications can be made thereto within the scope of the claims that follow. Accordingly, it is not intended that the scope of the invention in any way be limited by the above description, but instead be determined entirely by reference to the claims that follow.

The invention in which an exclusive right is claimed is defined by the following:

1. A method for indicating activeness of a clip to a user who is composing and/or editing a multimedia work that includes a plurality of clips, comprising the steps of:

(a) displaying a timeline for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating a plurality of tracks for a time interval corresponding to that represented by the timeline;

(b) enabling the user to select a clip as active, indicating that said clip is to contribute to an output comprising the multimedia work; and

(c) marking the clip selected by the user as active with a visual indicator, so that by inspection of the visual indicator and a position of the clip on one of the plurality of tracks, the user can determine a time during which the clip contributes to a content of the multimedia work.

2. The method of claim 1, further comprising the steps of enabling the user to select a portion of the clip as active; and,

marking the portion of the clip selected as active with the visual indicator, the position of the visual indicator relative to the clip indicating the portion of the clip that is active and contributes to the multimedia work.

3. The method of claim 2, wherein each clip is represented by a rectangle disposed on one of the timelines, a duration of said clip being indicated by the a relative size of the rectangle in at least one dimension, said visual indicator comprising a colored bar disposed adjacent one edge of the rectangle, a length of the colored bar and position of the colored bar relative to the rectangle indicating the duration of the portion of the clip that is active.

4. The method of claim 2, further comprising the step of automatically making active any portion of a clip added to one of the timelines, where said portion does not overlap any portion of another clip that is already active.

5. The method of claim 1, further comprising the steps of enabling the user to select two clips to be simultaneously active at one time when one of a transition and an overlay is to occur between the two clips; and, providing a visual indicator of the duration of said one of the transition and overlay between the two clips.

6. The method of claim 5, wherein the visual indicator comprises a color bar having a color that indicates a specific type of said one of the transition and the overlay between the clips.

7. The method of claim 1, wherein the user can visually perceive the contents of a plurality of clips that are included on the timeline to facilitate determining whether to make a clip active or inactive.

8. A system for indicating activeness of a clip to a user composing and/or editing a multimedia work that includes a plurality of clips, comprising:

(a) a monitor for displaying the multimedia work and for displaying screens used to compose and edit the multimedia work;

(b) a memory for storing a plurality of machine instructions defining a software program for composing and editing the multimedia work; and

(c) a processor, coupled to the memory and to the monitor, for executing the machine instructions stored in the memory, said machine instructions causing the processor to implement the following functions:

(i) display a timeline for at least a portion of the multimedia work on the monitor, said timeline graphically representing the portion of the multimedia work and graphically illustrating a plurality of tracks for a time interval corresponding to that represented by the timeline;

(ii) enable the user to make a clip as active, indicating that said clip does not contribute to an output comprising the multimedia work, or inactive, indicating that said clip does not contribute to said output; and

(iii) mark the clip selected by the user as active with a visual indicator shown on the monitor, so that by inspection of the visual indicator and a position of the clip on one of the plurality of tracks that are visible on the monitor, the user can determine a time during which the clip contributes to the output comprising the multimedia work.

9. The system of claim 8, wherein the machines instructions further cause the processor to:

(a) enable the user to select a portion of the clip as active; and

(b) mark the portion of the clip selected as active with the visual indicator, the position of the visual indicator relative to the clip indicating the portion of the clip that is active.

10. The system of claim 9, wherein each clip is represented by a rectangle disposed on one of the timelines, a duration of said clip being indicated by a relative size of at least one dimension of the rectangle on the monitor, said visual indicator comprising a colored bar disposed adjacent one edge of the rectangle, a length of the colored bar and position of the colored bar relative to the rectangle indicating the duration of the portion of the clip that is active.

11. The system of claim 9, wherein the machines instructions further cause the processor to automatically make active any portion of a clip added to one of the timelines that does not overlap any portion of another clip that is already active.

12. The system of claim 9, wherein the machines instructions further cause the processor to:

(a) enable the user to select two clips to be simultaneously active at one time when one of a transition and an overlay is to occur between the two clips; and

(b) provide a visual indicator of the duration of said one of the transition and overlay between the two clips.

13. The system of claim 12, wherein the visual indicator comprises a color bar.

14. The system of claim 12, wherein the color bar has a color that indicates a specific type of said one of the transition and the overlay between the clips.

15. A method for indicating effects applied to a specific portion of a multimedia work when composing and/or editing the multimedia work, said method comprising the steps of:

(a) displaying a timeline for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating at least one track for a time interval corresponding to that represented by the timeline;

(b) enabling the user to select an effect that will apply to the multimedia work and to select a time interval over which the effect will be applied, said effect being associated with a selected track of the timeline by the user and being applied to only a selected portion of the track; and

(c) providing an effect indicator that indicates the portion of the track to which the effect applies and the time interval of the effect.

16. The method of claim 15, wherein the track includes an effect region that extends adjacent to the track, said effect indicator being disposed within the track region.

17. The method of claim 15, wherein the effect indicator comprises a colored bar, a length of the colored bar and a position of the color bar relative to the track indicating the time interval over which the effect is applied.

18. The method of claim 17, wherein the effect is applied to a portion of any clip positioned on the track with which the effect is associated, adjacent the colored bar, but is otherwise independent of any clip.

19. The method of claim 17, further comprising the step of enabling the user to modify the time interval over which the effect is applied, including modifying a point in time at which the effect is applied and its duration.

20. The method of claim 15, further comprising the step enabling the user to associate a plurality of effects with the track, each effect having a user selectable time interval that is independent of the time interval of other effects associated with the track.

21. The method of claim 20, wherein the plurality of effects are represented by corresponding color bars that are disposed adjacent to the track with which the plurality of

effects are associated, a length and a position of each of the color bars indicating the time and duration at which each is applied to the track.

22. The method of claim 21, wherein predefined different colors are used for the color bars to represent different classes of effects, each predefined color representing a different class of effect.

23. The method of claim 21, wherein an order in which the plurality of effects are applied determines a result in the multimedia work, said color bars representing the plurality of effects being positioned relative to each other so as to indicate the order in which the effects are applied.

24. A system for indicating effects applied to a specific portion of a multimedia work when composing and/or editing the multimedia work, comprising:

(a) a monitor for displaying the multimedia work and for displaying screens used to compose and edit the multimedia work;

(b) a memory for storing a plurality of machine instructions defining a software program for composing and editing the multimedia work; and

(c) a processor, coupled to the memory and to the monitor, for executing the machine instructions stored in the memory, said machine instructions causing the processor to implement the following functions:

(i) display a timeline for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating at least one track for a time interval corresponding to that represented by the timeline;

(ii) enable the user to select an effect that will apply to the multimedia work and to select a time interval over which the effect will be applied, said effect being associated with a selected track of the timeline by the user and being applied to only a selected portion of the track; and

(iii) provide an effect indicator that indicates the portion of the track to which the effect applies and the time interval of the effect.

25. The system of claim 24, wherein the track with which the effect is associated includes an effect region that extends adjacent to the track, said effect indicator being disposed within the track region.

26. The system of claim 24, wherein the effect indicator comprises a colored bar, a length of the colored bar and a position of the color bar relative to the track indicating the time interval over which the effect is applied.

27. The system of claim 24, wherein the effect is applied to a portion of any clip positioned on the track with which the effect is associated, adjacent the colored bar, but is otherwise independent of the clip.

28. The system of claim 24, wherein the machine instructions further cause the processor to enable the user to modify the time interval over which the effect is applied, including enabling the user to modify a point in time at which the effect is applied and its duration.

29. The system of claim 24, wherein the machine instructions further cause the processor to enable the user to associate a plurality of effects with the track, each effect having a user selectable time interval that is independent of the time interval of other effects associated with the track.

30. The system of claim 29, wherein different colors are used for the color bars to represent different classes of effects.

31. The system of claim 29, wherein an order in which the plurality of effects are applied determines a result in the multimedia work, said color bars representing the plurality of effects being positioned relative to each other so as to indicate the order in which the effects are applied.

32. A method for indicating a plurality of effects that are applied to clips comprising a multimedia work, when composing and/or editing the multimedia work, said method comprising the steps of:

(a) displaying a timeline for at least a portion of the multimedia work, said timeline graphically representing a portion of the multimedia work and graphically illustrating at least one track on which is disposed at least one clip;

(b) enabling the user to select a plurality of different effects that will apply to said clip of the multimedia work and to select a portion of the clip to which each effect will be applied; and

(c) providing an effect indicator for each effect that indicates the portion of the clip to which the effect applies and the time interval during which the effect applies.

33. The method of claim 32, wherein the effect indicator for each clip comprises a colored bar, a length of the bar and its relative position adjacent the clip indicating the portion of the clip to which the effect applies.

34. The method of claim 33, wherein the colored bars representing each of the plurality of effects selected by the user for application to the clip are stacked adjacent to the clip.

35. The method of claim 34, wherein an order in which the colored bars are stacked indicates a sequence in which the effects are applied to the clip to produce a result.

36. The method of claim 32, wherein the plurality of effects are associated with the clip to which the plurality of effects apply, so that moving the clip along the timeline causes movement of the effects applied to the clip.

37. The method of claim 32, further comprising the step of displaying a descriptive label when a cursor is moved over any effect indicator, said descriptive label identifying a class of effect that is represented by the effect indicator.

38. A system for indicating a plurality of effects that are applied to clips comprising a multimedia work, when composing and/or editing the multimedia work, comprising:

(a) a monitor for displaying the multimedia work and for displaying screens used to compose and edit the multimedia work;

(b) a memory for storing a plurality of machine instructions defining a software program for composing and editing the multimedia work; and

(c) a processor, coupled to the memory and to the monitor, for executing the machine instructions stored in the memory, said machine instructions causing the processor to implement the following functions:

(i) display a timeline on the monitor for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating at least one track on which is represented at least one clip;

(ii) enable the user to select a plurality of different effects that will apply to said clip of the multimedia work and to select a portion of the clip to which each effect will be applied; and

(iii) provide an effect indicator to be displayed on the timeline that indicates the portion of the clip to which the effect applies and the time interval within that of the clip, during which the effect applies.

39. The system of claim 38, wherein the effect indicator for each clip comprises a colored bar, a length of the bar and

its relative position adjacent the clip indicating the portion of the clip to which the effect applies.

40. The system of claim 39, wherein the colored bars representing each of the plurality of effects selected by the user for application to the clip are stacked adjacent to the clip.

41. The system of claim 39, wherein an order in which the colored bars are stacked indicates a sequence in which the effects are applied to the clip to produce a result.

42. The system of claim 38, wherein the plurality of effects are associated with the clip to which the plurality of effects apply, so that movement of the clip along the timeline causes movement of the effects applied to the clip.

43. The system of claim 38, wherein the machine instructions further cause the process to display a descriptive label on the monitor when a cursor is moved over any effect indicator, said descriptive label identifying a class of effect that is represented by the effect indicator.

44. An article of manufacture adapted to be used with a computer for indicating activeness of a clip to a user who is composing and/or editing a multimedia work that includes a plurality of clips, comprising:

(a) a nonvolatile memory device; and

(b) a plurality of machine instructions stored on the nonvolatile memory device, said machine instructions, when executed by the computer, causing it to:

(i) display a timeline for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating a plurality of tracks for a time interval during the multimedia work corresponding to that represented by the timeline;

(ii) enable the user to make a clip active, indicating that said clip is to contribute to an output comprising the multimedia work, or inactive, indicating that said clip does not contribute said output; and

(iii) mark the clip selected by the user as active with a visual indicator, so that by inspection of the visual indicator and a position of the clip on one of the plurality of tracks, the user can determine a time during which the clip contributes to a content of the multimedia work.

45. An article of manufacture adapted to be used with a computer for indicating effects applied to a specific portion of a multimedia work when composing and/or editing the multimedia work, comprising:

(a) a nonvolatile memory device; and

(b) a plurality of machine instructions stored on the nonvolatile memory device, said machine instructions, when executed by the computer, causing it to:

(i) display a timeline for at least a portion of the multimedia work, said timeline graphically representing the portion of the multimedia work and graphically illustrating at least one track for a time interval corresponding to that represented by the timeline;

(ii) enable the user to select an effect that will apply to the multimedia work and to select a time interval over which the effect will be applied, said effect being associated with a selected track of the timeline by the user and being applied to only a selected portion of the track; and

(iii) provide an effect indicator that indicates the portion of the track to which the effect applies and the time interval of the effect.

46. An article of manufacture adapted to be used with a computer for indicating a plurality of effects that are applied to clips comprising a multimedia work, when composing and/or editing the multimedia work, comprising:

(a) a nonvolatile memory device; and

(b) a plurality of machine instructions stored on the nonvolatile memory device, said machine instructions, when executed by the computer, causing it to:

(i) display a timeline for at least a portion of the multimedia work, said timeline graphically representing a portion of the multimedia work and graphically illustrating at least one track on which is disposed at least one clip;

(ii) enable the user to select a plurality of different effects that will apply to said clip of the multimedia work and to select a portion of the clip to which each effect will be applied; and

(iii) provide an effect indicator for each effect that indicates the portion of the clip to which the effect applies and the time interval during which the effect applies.

* * * * *

US005889519A

United States Patent [19]

Boezeman et al.

[11] Patent Number: 5,889,519

[45] Date of Patent: *Mar. 30, 1999

[54] METHOD AND SYSTEM FOR A MULTIMEDIA APPLICATION DEVELOPMENT SEQUENCE EDITOR USING A WRAP CORRAL

[75] Inventors: John Junior Boezeman; Dennis Donald King, both of Cary; Christopher Joseph Paul, Durham, all of N.C.

[73] Assignee: International Business Machines Corp., Armonk, N.Y.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: 630,866

[22] Filed: Mar. 26, 1996

[51] Int. Cl.[6] G06F 15/00

[52] U.S. Cl. 345/340; 345/342

[58] Field of Search 345/340, 326–356; 395/342, 341, 343, 346, 375, 339, 418, 133

[56] References Cited

U.S. PATENT DOCUMENTS

5,165,012	11/1992	Crandall et al.	395/347
5,388,197	2/1995	Rayner	395/328
5,404,316	4/1995	Klingler et al.	364/514
5,416,900	5/1995	Blanchard	395/346
5,448,695	9/1995	Douglas et al.	395/347
5,519,828	5/1996	Rayner	395/340
5,548,702	8/1996	Li et al.	395/348
5,553,222	9/1996	Milne et al.	395/340
5,619,639	4/1997	Mast	345/326
5,630,080	5/1997	Malemud et al.	395/346
5,680,535	10/1997	Harbin et al.	345/473
5,689,484	11/1997	Hirasawa	369/33

Primary Examiner—Steven P. Sax

Attorney, Agent, or Firm—A. Bruce Clay

[57] ABSTRACT

Parts for a multimedia title may be extended indefinitely relative to time with a user interface tool. A part for a multimedia title is provided with edges having a wrap handle affixed to one edge. By dragging and dropping the wrap handle to a wrap corral, the part will play or be visible indefinitely.

6 Claims, 16 Drawing Sheets

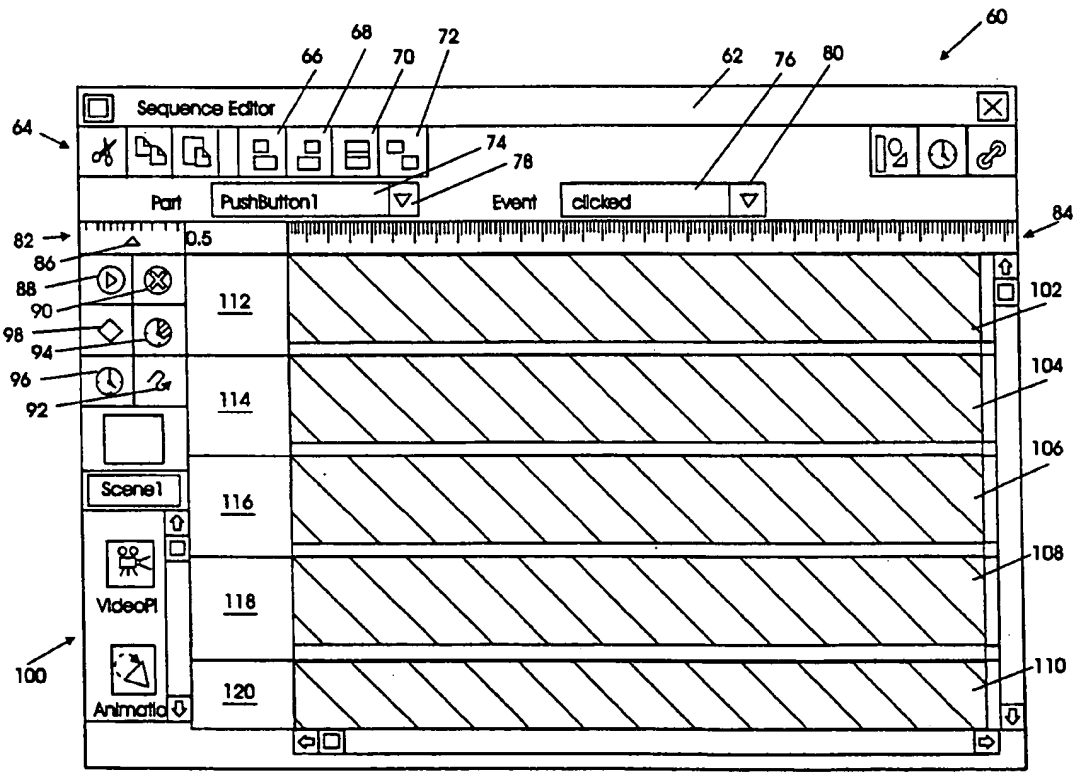
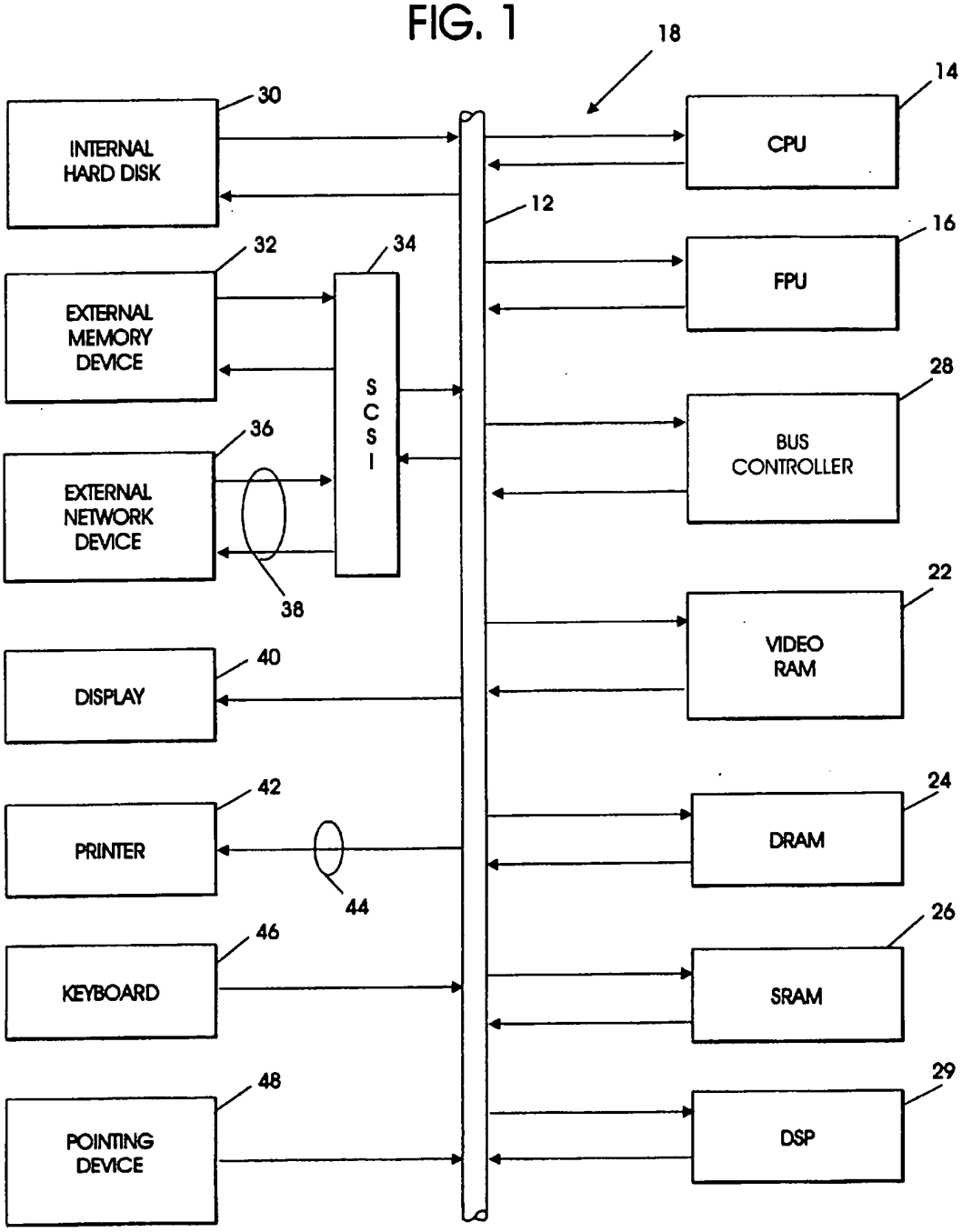


FIG. 1


18
INTERNAL HARD DISK
30
CPU
14
12
EXTERNAL MEMORY DEVICE
32
34
FPU
16
S C S I
EXTERNAL NETWORK DEVICE
36
38
BUS CONTROLLER
28
VIDEO RAM
22
DISPLAY
40
PRINTER
42
44
DRAM
24
KEYBOARD
46
SRAM
26
POINTING DEVICE
48
DSP
29

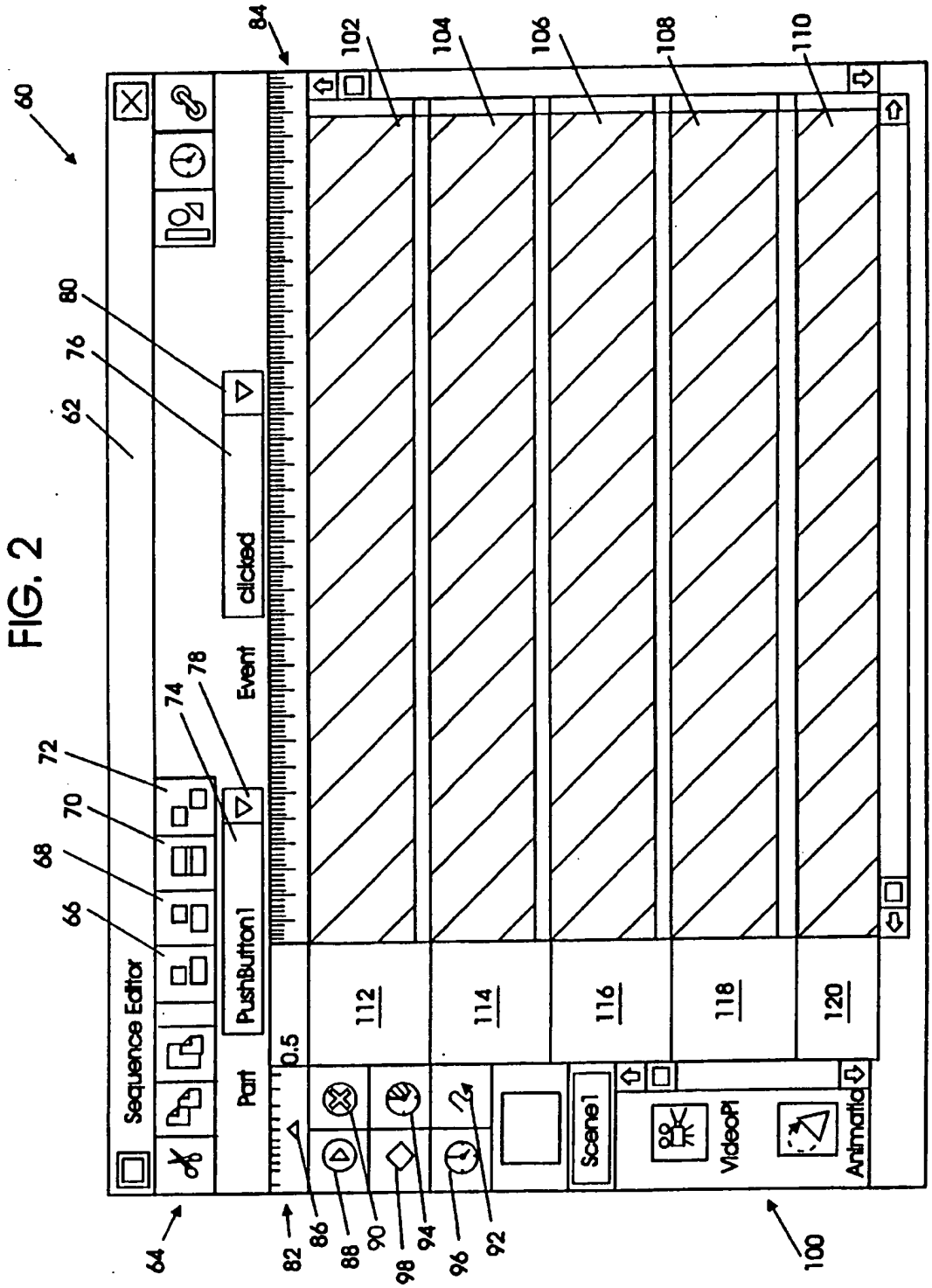

60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100
102
104
106
108
110
112
114
116
118
120
Sequence Editor
Part
PushButton1
Event
clicked
0.5
Scene1
VideoP1
Animatio

FIG. 2

FIG. 3

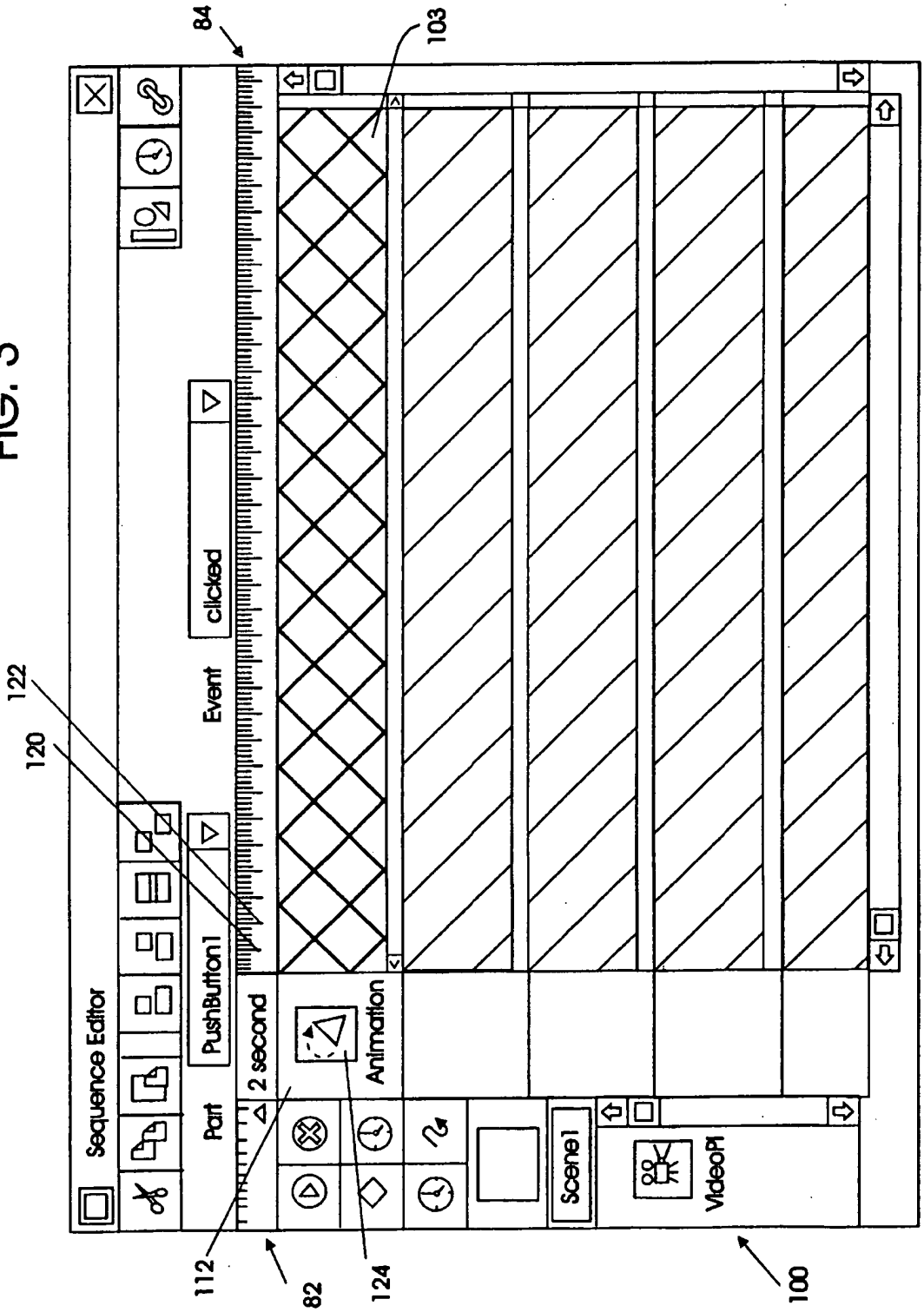

84
103
122
120
Event
clicked
PushButton1
Sequence Editor
Part
2 second
Animation
Scene1
VideoPl
112
82
124
100

FIG. 4

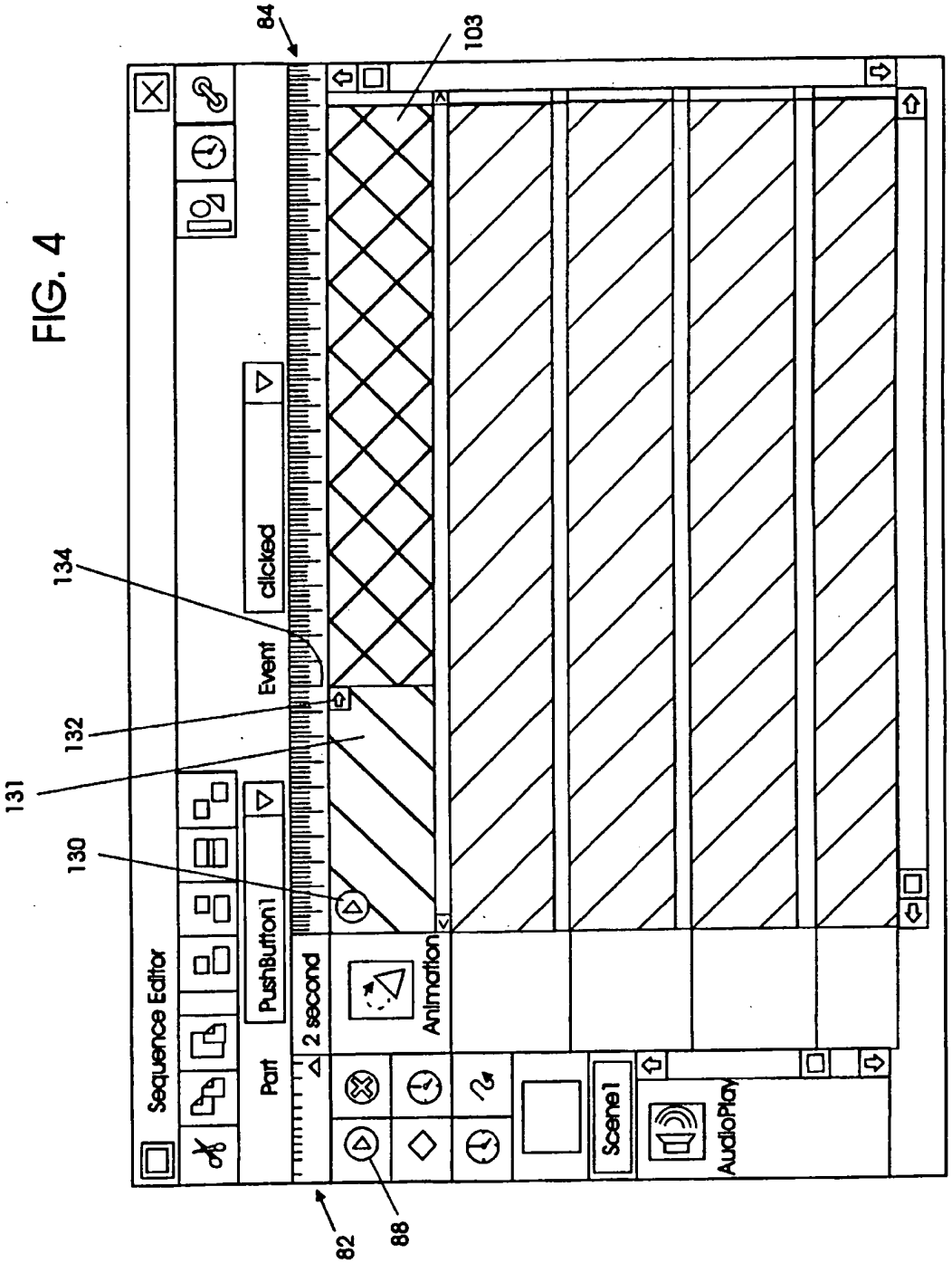

84
103
134
132
131
130
clicked
Event
PushButton1
Part
2 second
Animation
Sequence Editor
Scene1
AudioPlay
82
88

FIG. 5

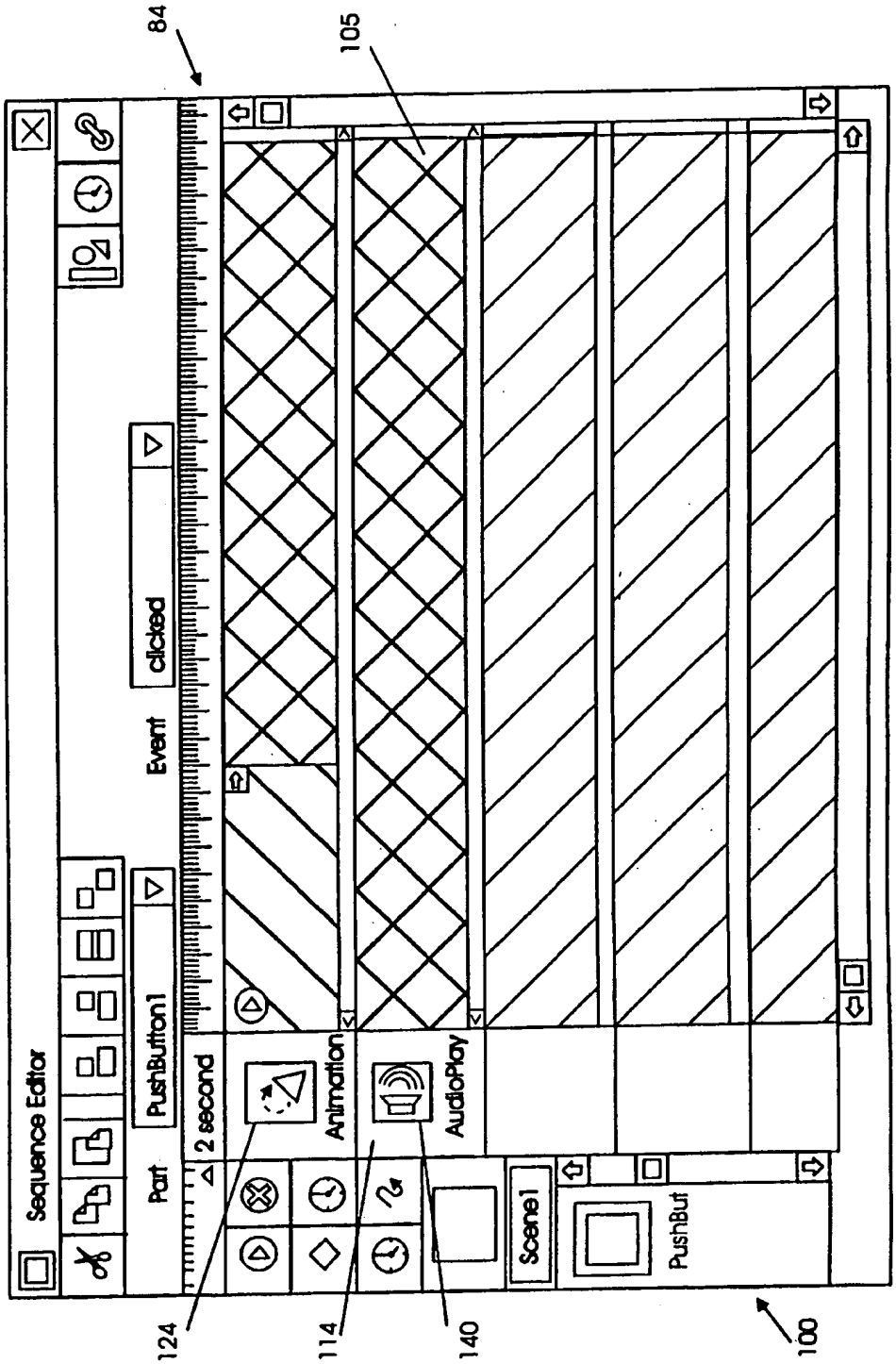

84
105
clicked
Event
PushButton1
Part
2 second
Animation
AudioPlay
Sequence Editor
Scene1
PushBut
124
114
140
100

FIG. 6

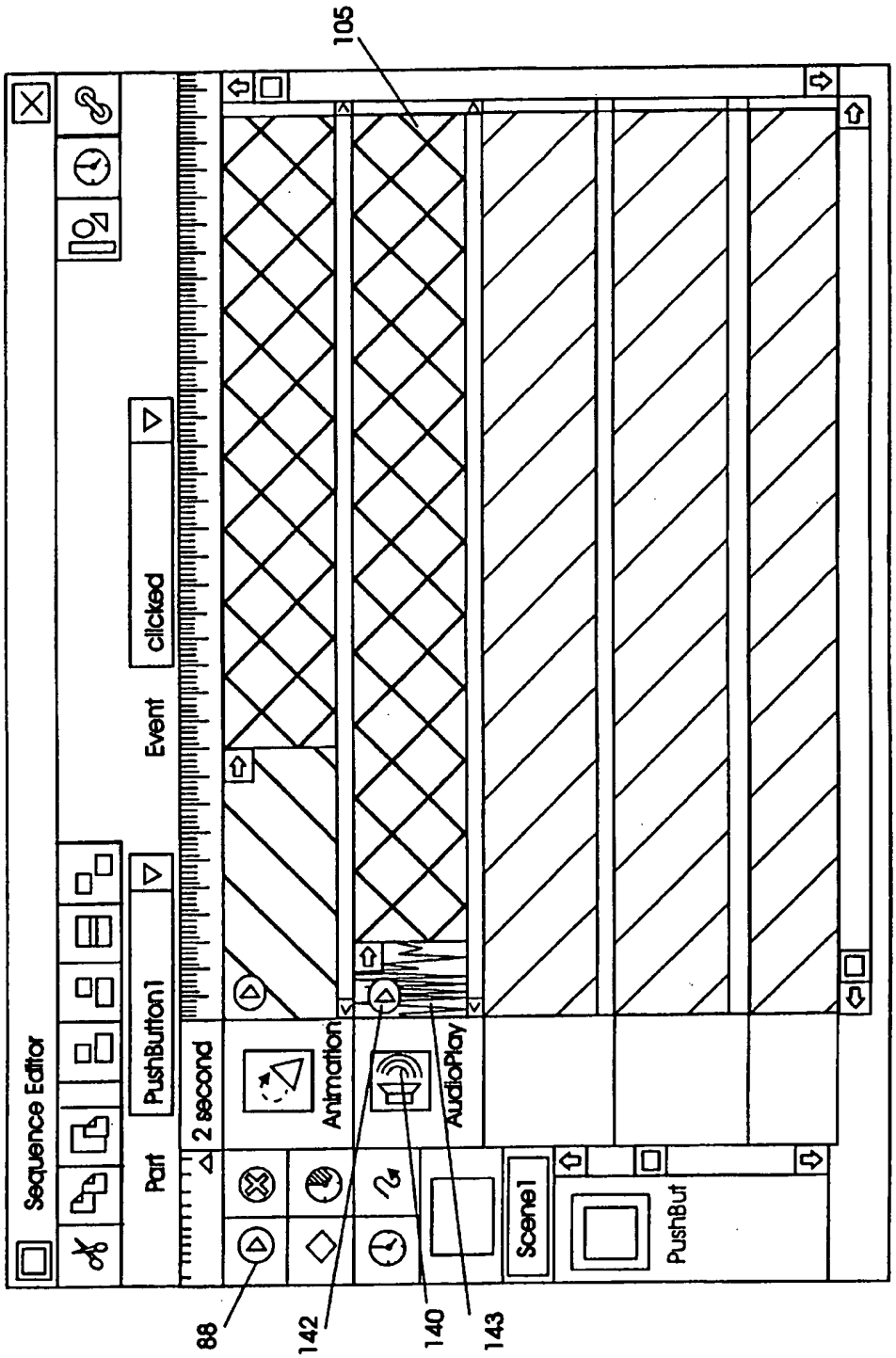

105
Sequence Editor
Part
PushButton1
Event
clicked
2 second
Animation
AudioPlay
Scene1
PushBut
88
142
140
143

FIG. 7

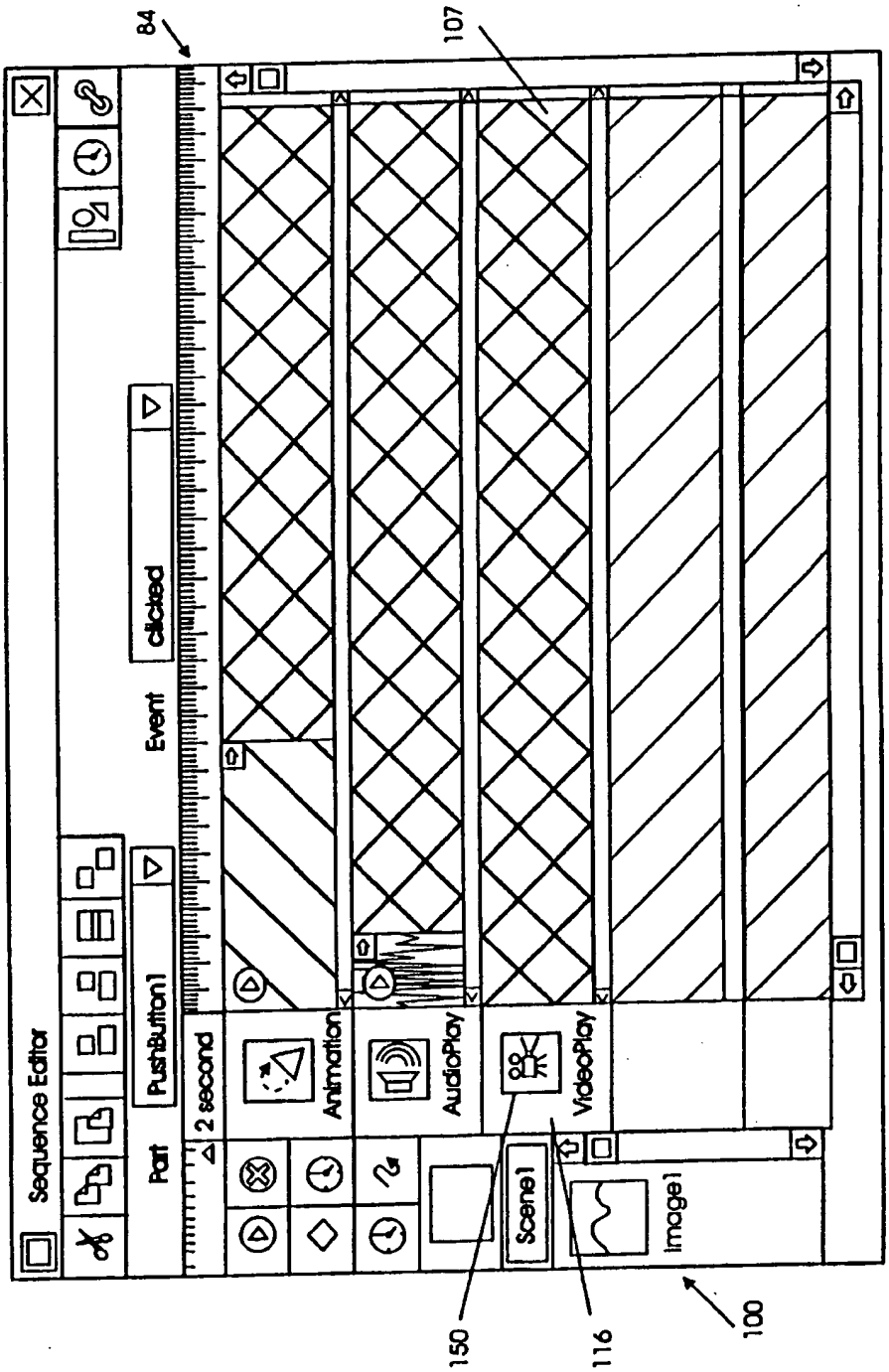

84
107
Event
clicked
PushButton1
Part
2 second
Sequence Editor
Animation
AudioPlay
VideoPlay
Scene1
Image1
150
116
100

FIG. 8

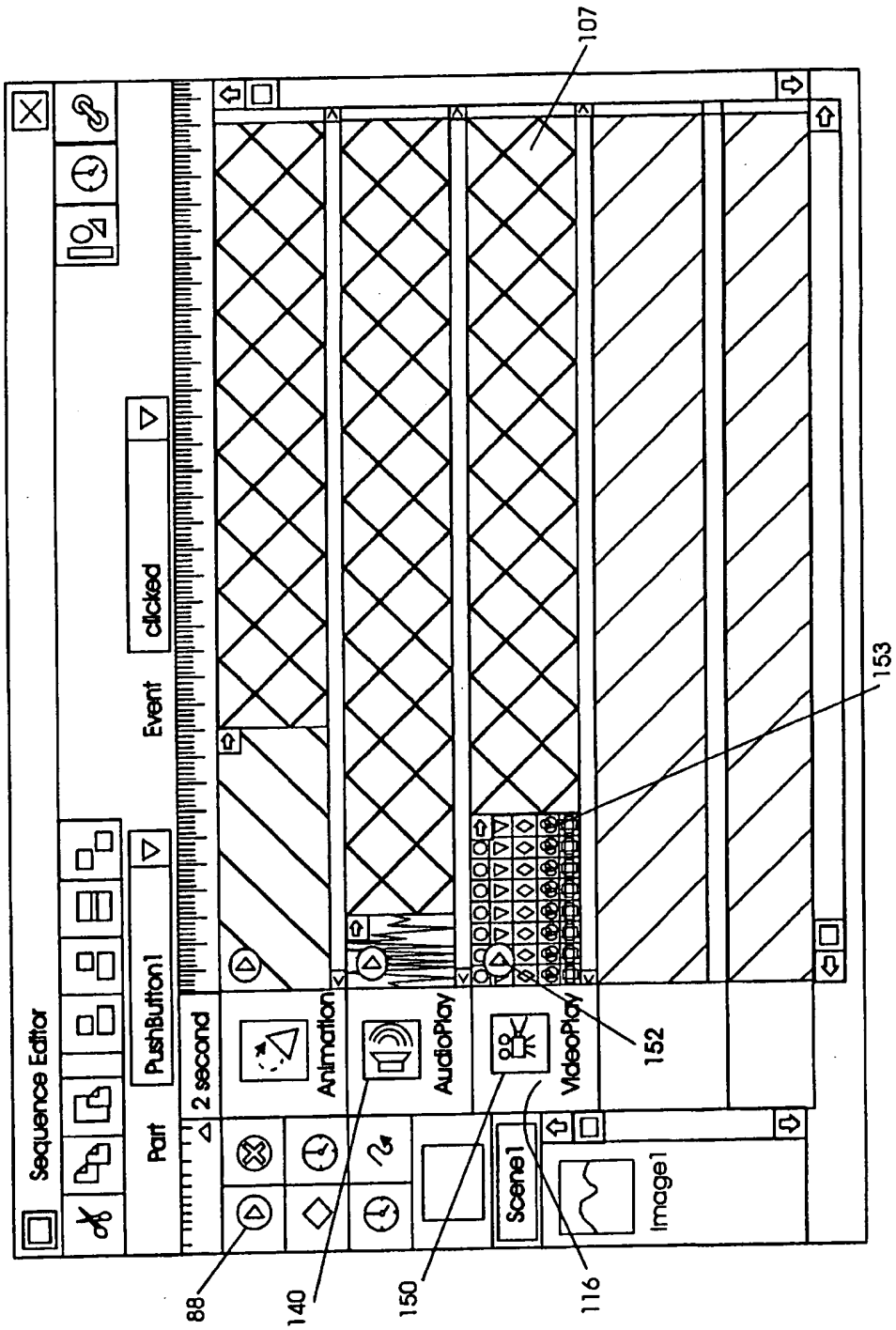

107
153
Sequence Editor
Part
PushButton1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
152
Scene1
Image1
88
140
150
116

FIG. 9

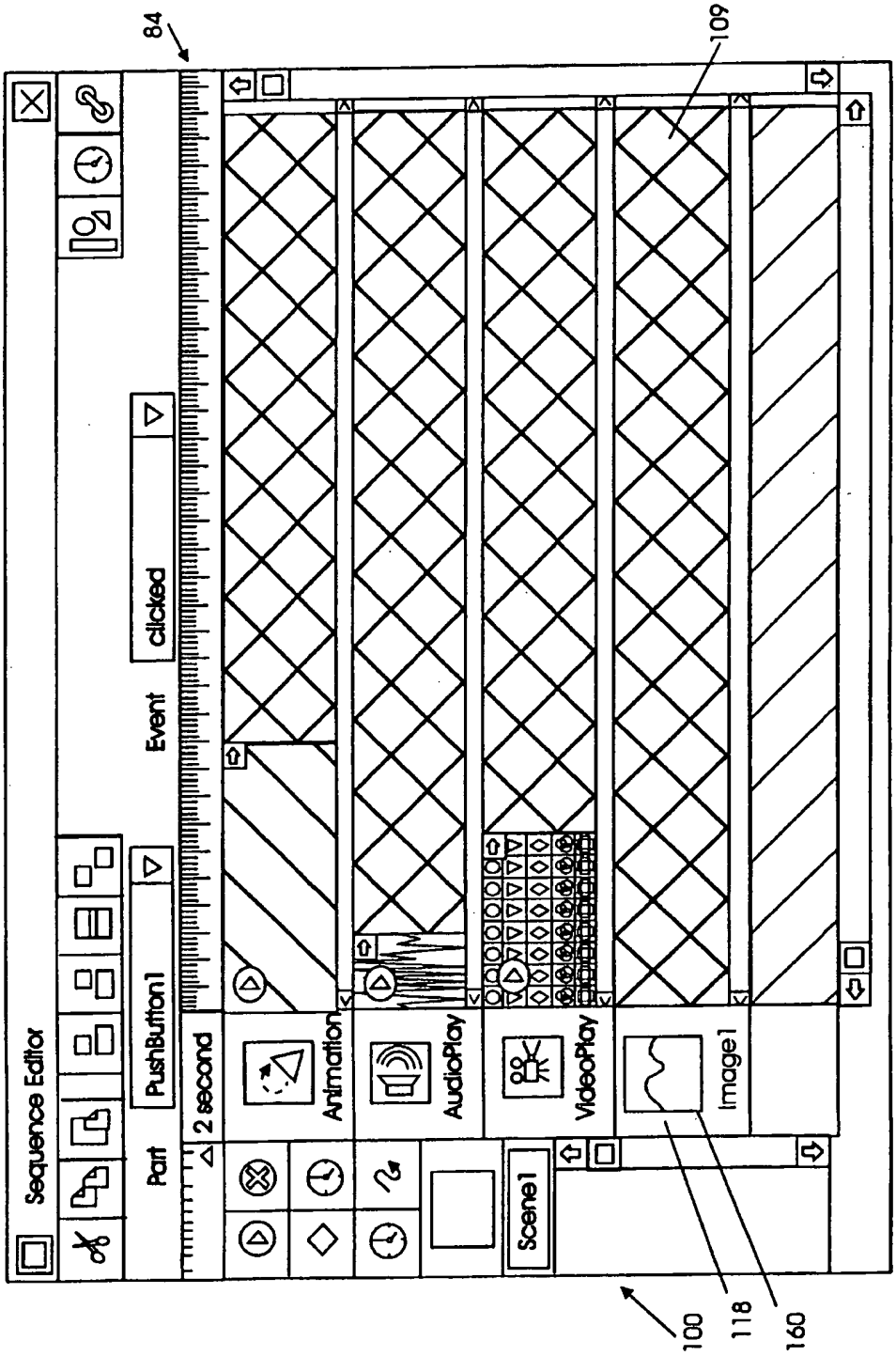

84
109
Sequence Editor
Part
PushButton1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1
100
118
160

FIG. 10

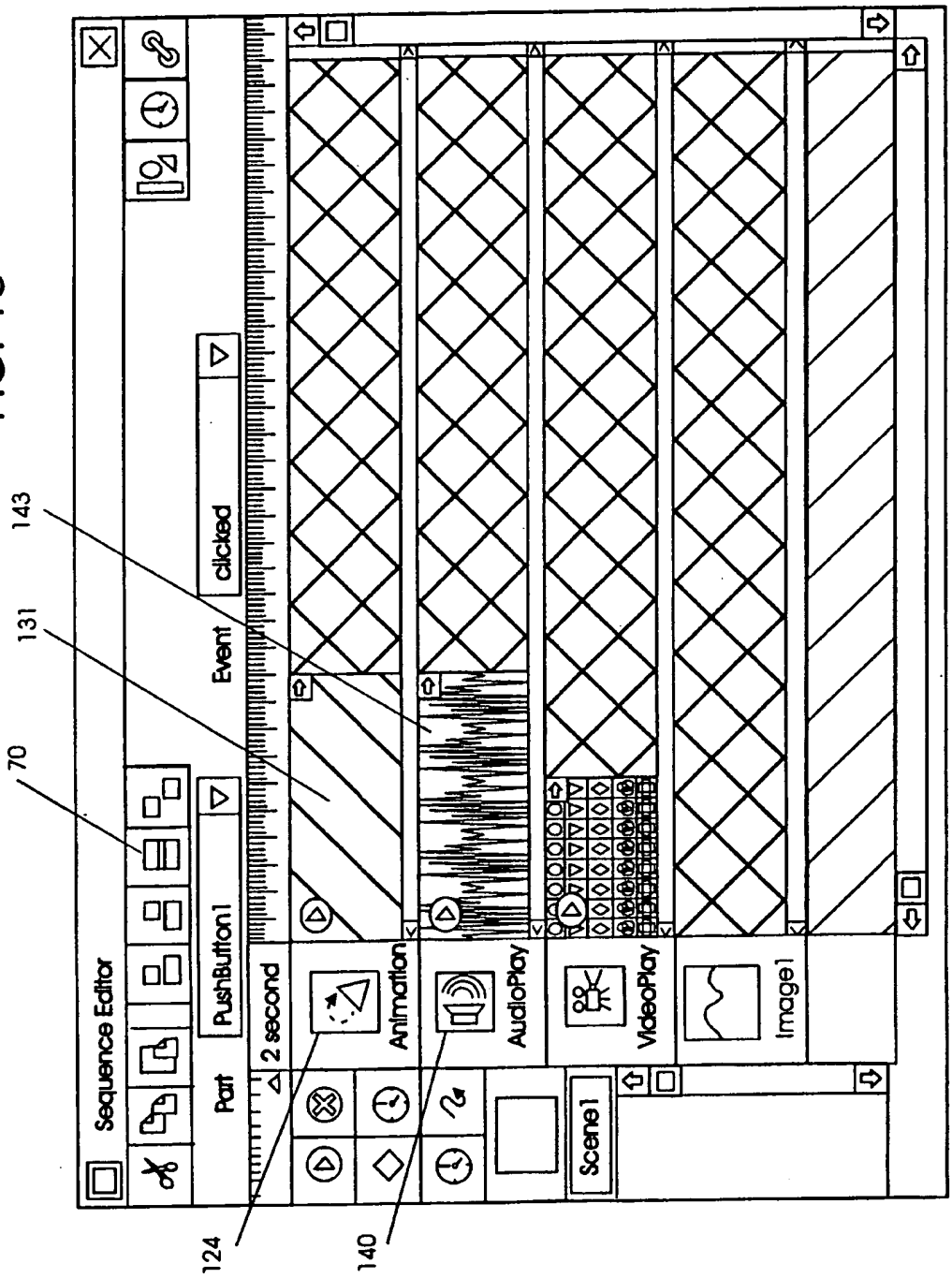

143
131
70
Event
clicked
PushButton1
Part
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1
Sequence Editor
124
140

FIG. 11

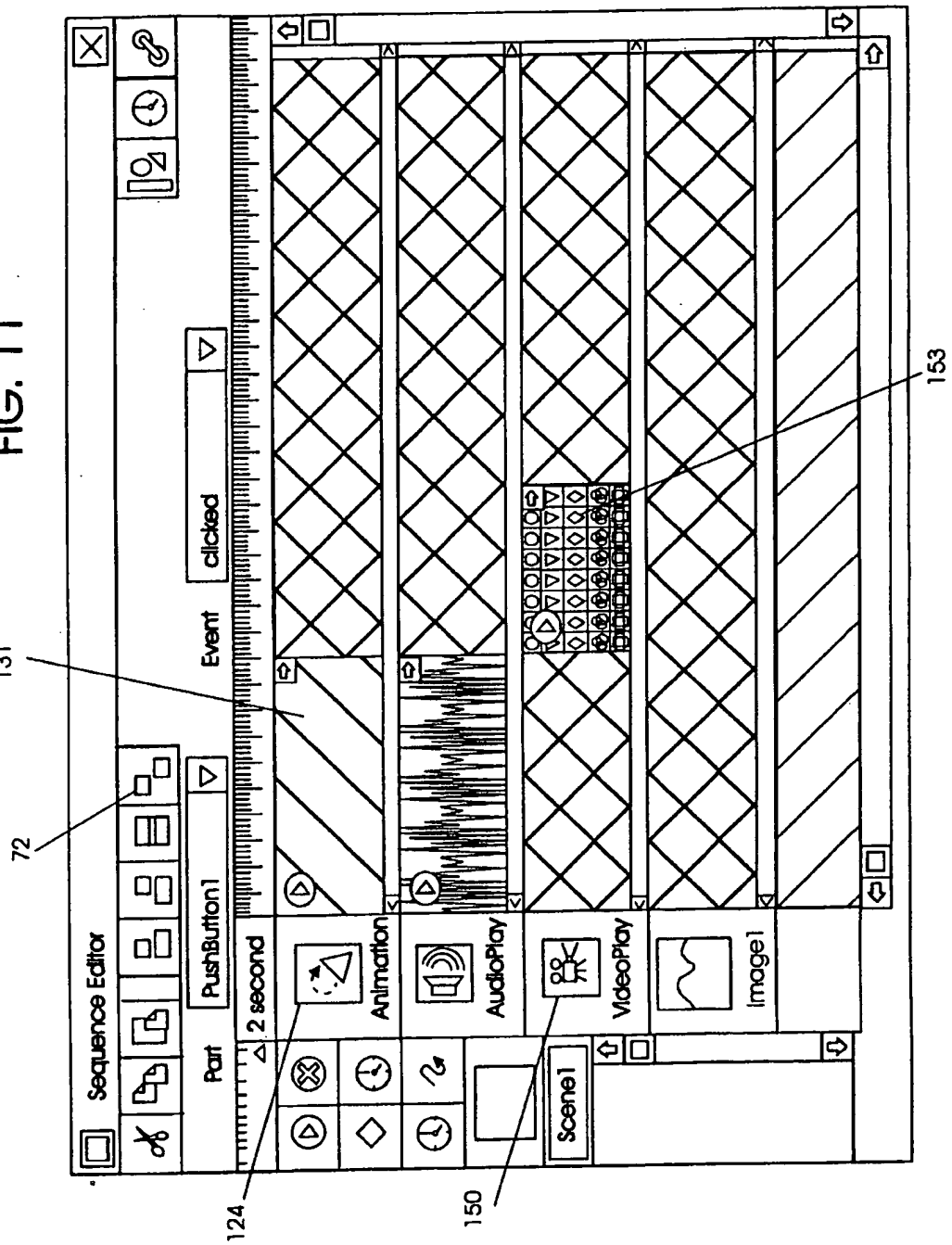

Sequence Editor
Part
PushButton1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1
131
72
153
124
150

FIG. 12

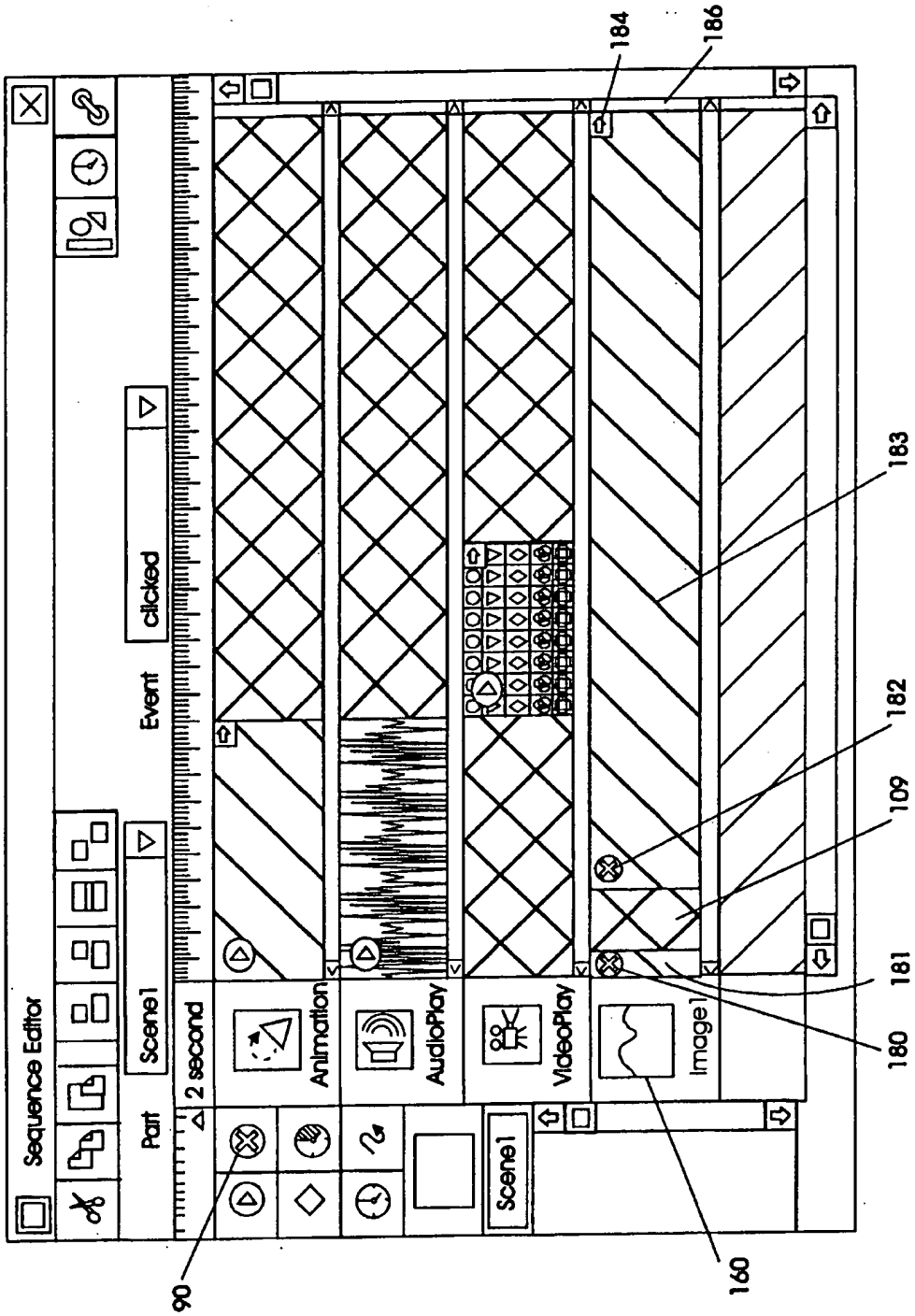

184
186
183
182
109
181
180
160
90
Sequence Editor
Part
Scene1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1

FIG. 13

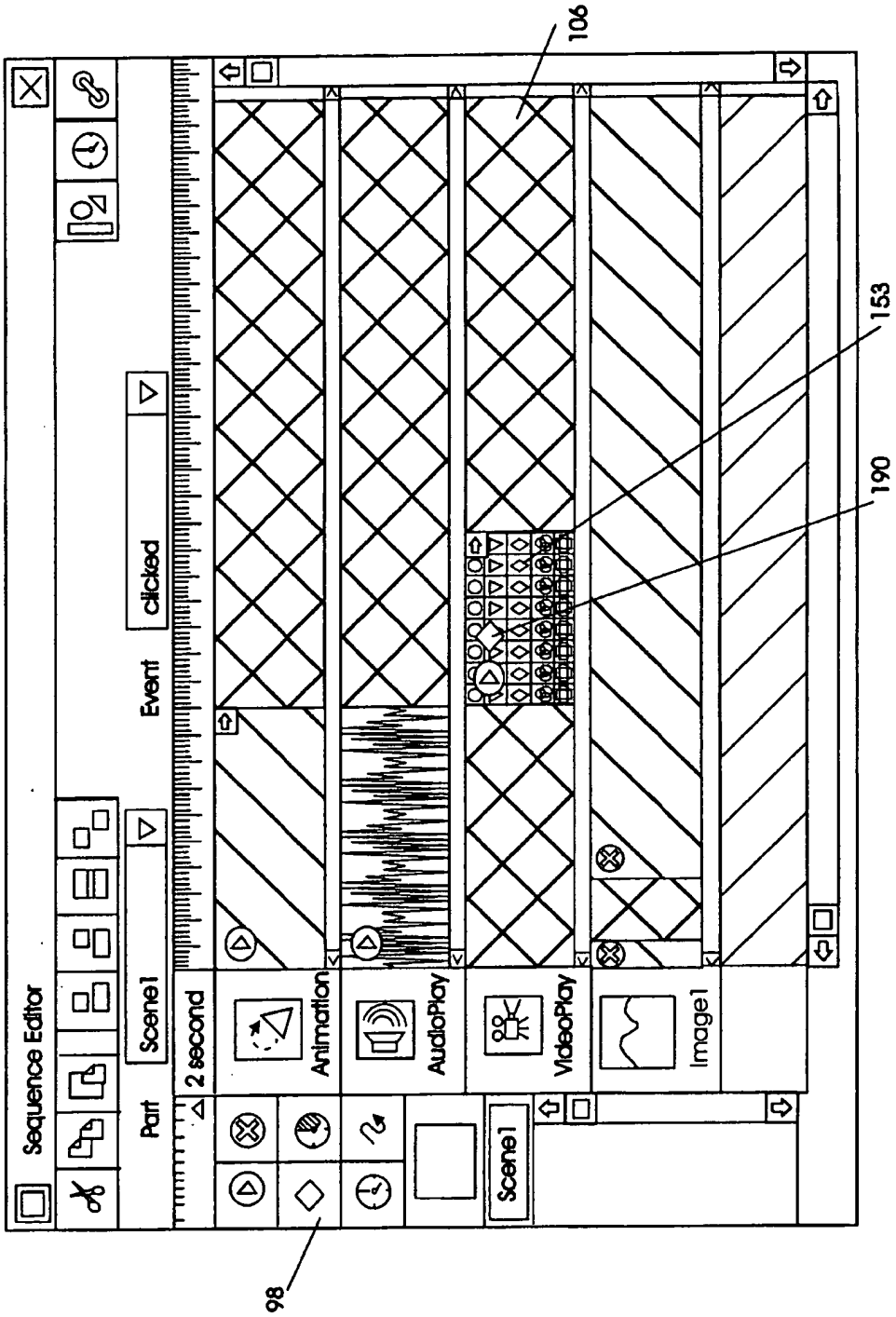

106
153
190
98
Sequence Editor
Part
Scene1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1

FIG. 14

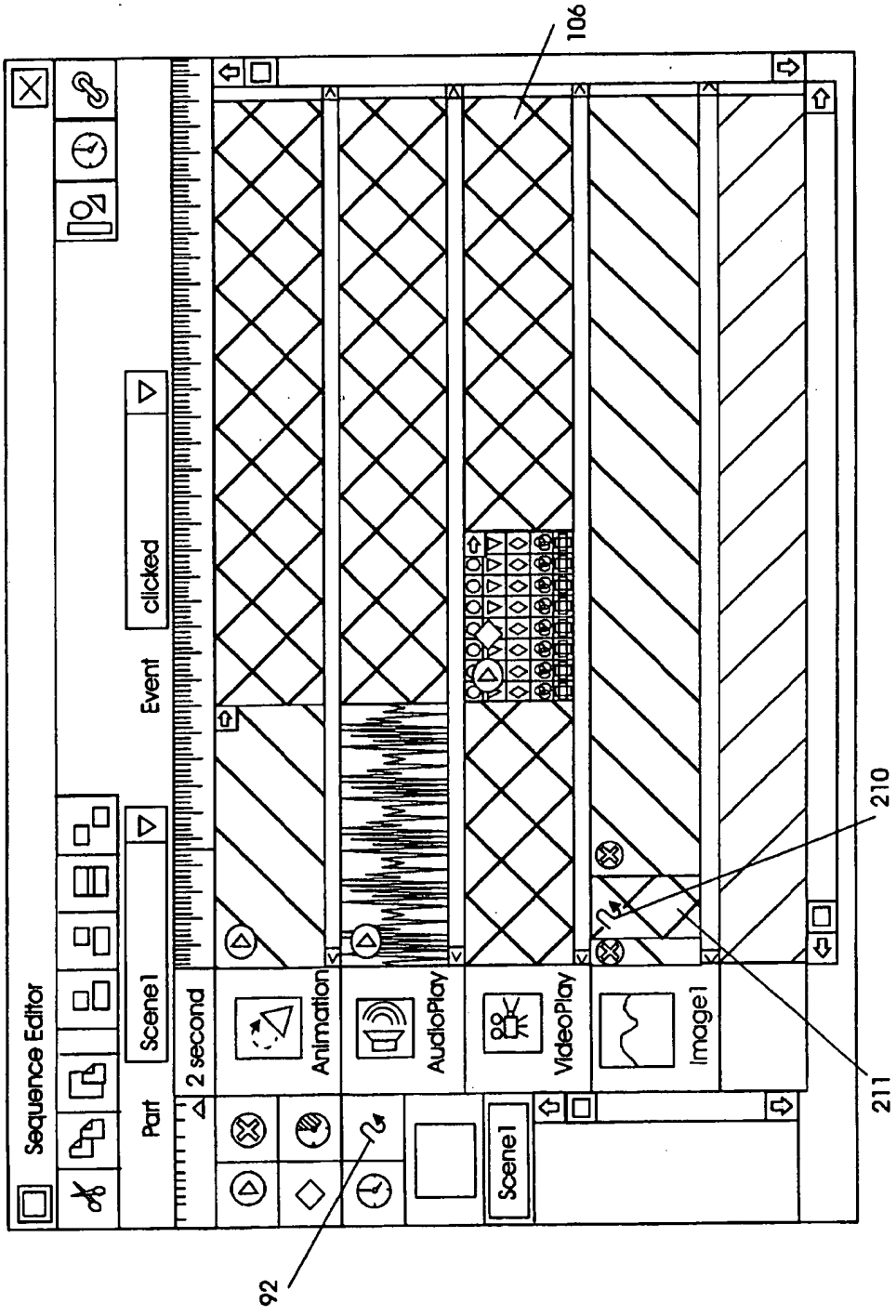
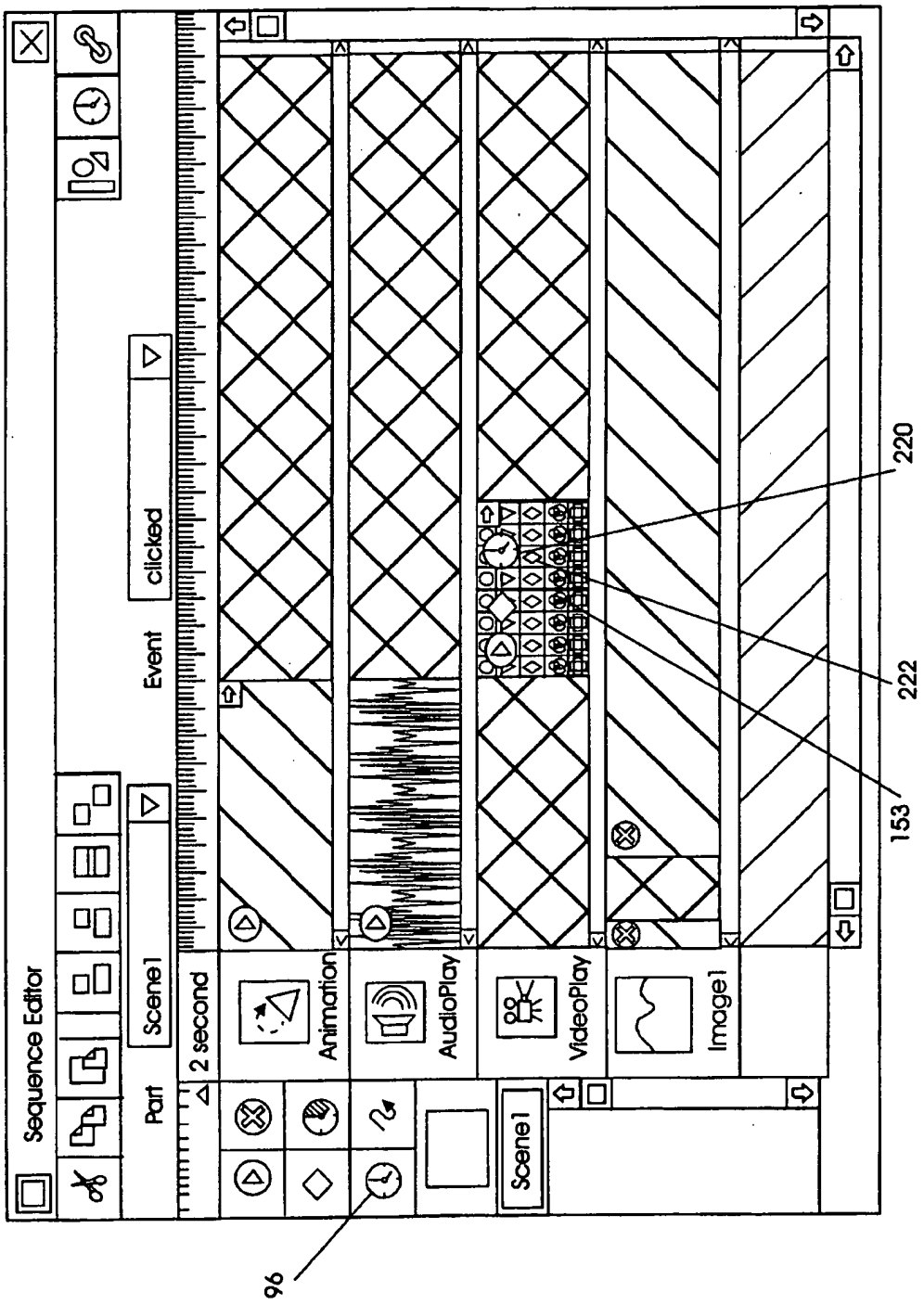

106
Sequence Editor
Part
Scene1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1
210
211
92

FIG. 15


Sequence Editor
Part
Scene1
Event
clicked
2 second
Animation
AudioPlay
VideoPlay
Image1
Scene1
220
222
153
96

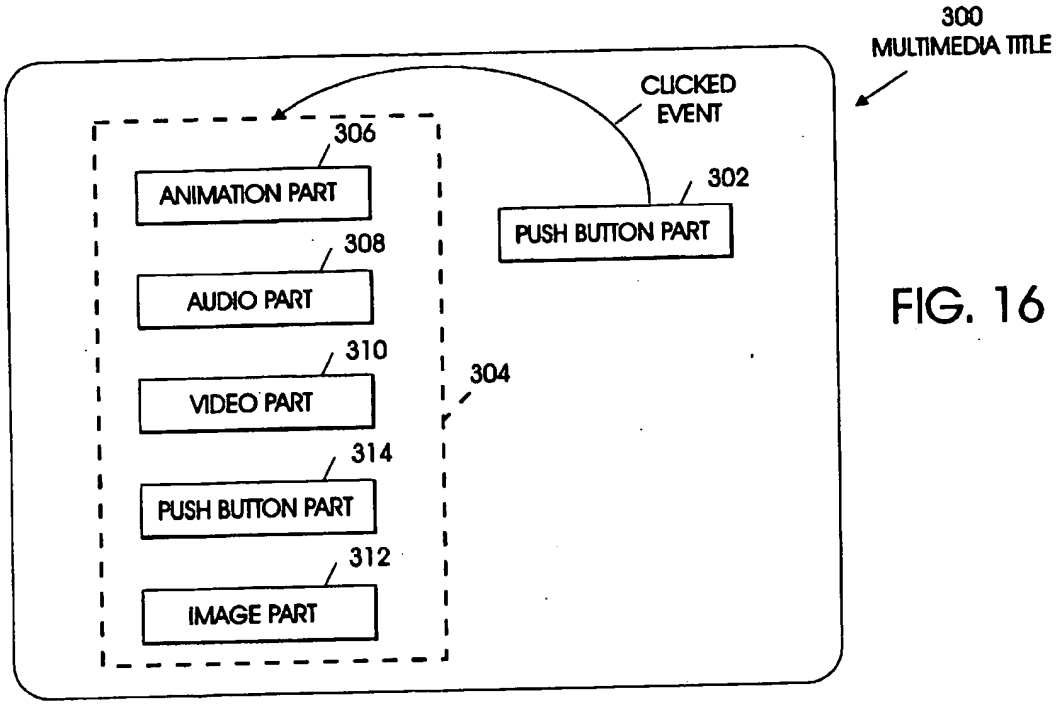

300
MULTIMEDIA TITLE
CLICKED EVENT
306
ANIMATION PART
308
AUDIO PART
310
VIDEO PART
314
PUSH BUTTON PART
312
IMAGE PART
304
302
PUSH BUTTON PART

FIG. 16

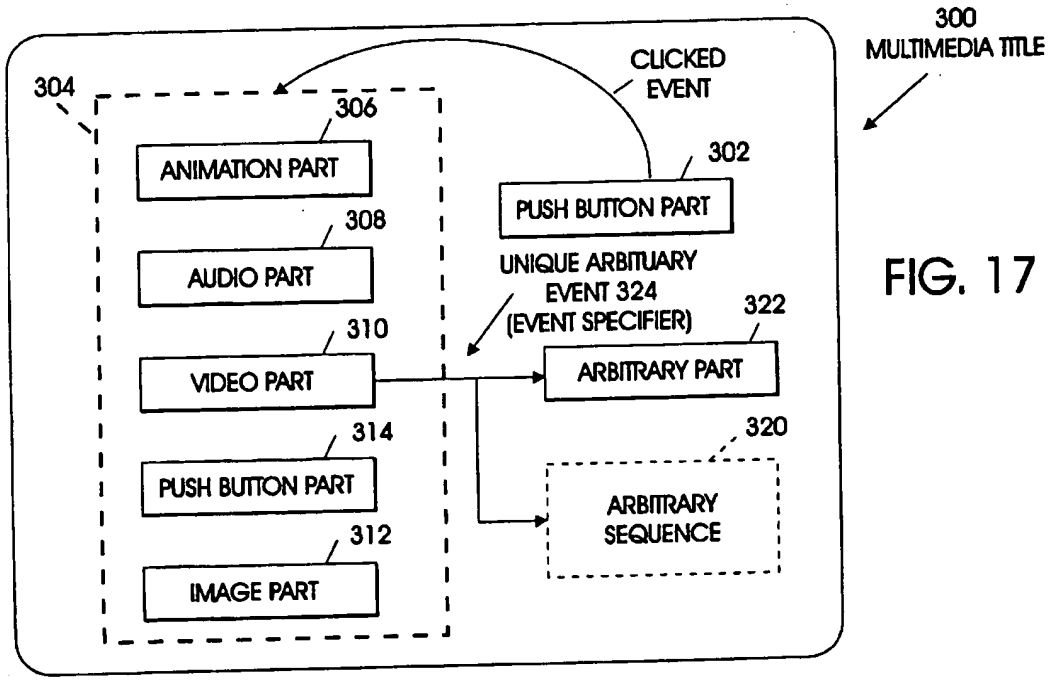

300
MULTIMEDIA TITLE
CLICKED EVENT
304
306
ANIMATION PART
308
AUDIO PART
310
VIDEO PART
314
PUSH BUTTON PART
312
IMAGE PART
302
PUSH BUTTON PART
UNIQUE ARBITUARY EVENT 324 (EVENT SPECIFIER)
322
ARBITRARY PART
320
ARBITRARY SEQUENCE

FIG. 17

METHOD AND SYSTEM FOR A MULTIMEDIA APPLICATION DEVELOPMENT SEQUENCE EDITOR USING A WRAP CORRAL

CROSS REFERENCE TO RELATED APPLICATIONS

This Application is related in subject matter to the following Applications filed concurrently herewith and assigned to a common Assignee:

Application Ser. No. 08/625,085 filed by King, et al., entitled "Method and System for a Multimedia Application Development Sequence Editor"; application Ser. No. 08/623,727 filed by King, et al., entitled "Method and System for a Multimedia Application Development Sequence Editor Using Spacer Tools"; application Ser. No. 08/625,084 filed by King, et al., entitled "Method and System for a Multimedia Application Development Sequence Editor Using Synchronizing Tools"; and application Ser. No. 08/622,909 filed by King, et al., entitled "Method and System for a Multimedia Application Development Sequence Editor Using Time Event Specifiers".

The foregoing co-pending Applications are incorporated by reference.

TECHNICAL FIELD OF THE INVENTION

The present invention relates in general to computer software, and in particular, to a method and system for the creation of multimedia titles with a user friendly sequence editor using a wrap corral.

BACKGROUND OF THE INVENTION

Multimedia is a term used in connection with computer systems to refer to software that processes, i.e., creates, edits, displays, synchronizes, etc., one or more files of time-based data. A multimedia computer system is designed to present various materials in various combinations of text, graphics, video, image, animation, audio, etc. Multimedia systems commonly comprise a combination of hardware and software including a personal computer, a CD-ROM drive, software editors, etc. The data for multimedia presentations are generally stored on hardware memory devices, such as a magnetic disc drive, a ROM compact disk player, video tape device, etc.

Application developers utilizing such a multimedia system may create programs that allow end users to view a multimedia presentation and manipulate various controls.

One of the most difficult problems that face creators/developers of multimedia titles is synchronization of time and events. For example, a developer must be able to synchronize audio with a video presentation and provide images that appear and disappear over time. In many complex real world examples it is desirable for two or more multimedia items to be synchronized based on one or more of the following: an event; a relative time relationship (i.e., A Before B, A After B, A coincident with B); and an absolute time.

One product attempting to deal with time synchronization is Director from Macromedia. Director specifies time on a frame based metaphor. A Score is used in Director to specify when things occur in a movie (or title). The Score is a sequence of frames that hold cells of an animation and the indication to start playing audio or video. Things are specified linearly in Director. Frames are sequenced in order unless software code is written to manually go back to a previous frame. Frames show sequentially one after the other. To create an animation, for example, a developer must manually associate each cell in the animation with a frame in the Score. There is no convenient way to cause things to occur in the relative manner. If two animations are to co-occur, corresponding frames must be manually specified on an individual basis. A Time tool can be used to specify the playing of a video; however, nothing else can be processed (i.e., animation, other videos, etc.) while the video is playing. There is no point and click mechanism for wrap; code must be written to cause a wrap.

Another product, Premier from Adobe, is a video editing product. It has a facility called the Construction Window (CW) where clips (i.e., video, scan images, QuickTime, movies, audio) are manipulated to create movies. The CW has channels, a timeline, and a time unit selector to adjust the scale of the ticks on the timeline. Pieces of video and images are put in channels and manually aligned (e.g. via dragging) with an absolute time and with other pieces of media. There is no notion of relative time or event time and no alignment helper co-tools. The notion of wrap does not exist for the creation of a movie.

Finally, IconAuthor (IA) is a multimedia authoring tool which allows creation of applications that include time-based media, logic, database, etc. IA uses a flowchart metaphor to specify an application. Icons representing the playing of media, database access, comparisons, assignment, etc. are taken from a palette and connected into a flowchart. Processing in the application then follows the hierarchical structure defined in the flowchart. There is no editor to synchronize processing visually. Synchronization is done by checking logic and by corresponding logic that responds to events.

Thus, there is a need for a method that simply and visually provides synchronization of multimedia parts in relation to event time, relative time, and absolute time and uses a wrap corral.

SUMMARY OF THE INVENTION

The present invention disclosed herein comprises a method and apparatus for extending parts of a multimedia title indefinitely relative to time which substantially eliminates problems associated with prior time extending programs. The present invention allows a user a simple and easy to use method for continuously playing or showing a part.

In accordance with one aspect of the invention, a user interface tool is provided for extending parts of a multimedia title indefinitely relative to time. A graphical representation of the part is provided with edges and a dragging handle positioned on at least one of the edges. A user may then simply grab the dragging handle and drag the edge into a wrap corral. Once the edge has been placed into the wrap corral, the part will play or be visible indefinitely.

It is a technical advantage of the present invention that a developer is no longer required to write code to replay or continuously show a multimedia part for a multimedia title.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and for further advantages thereof, reference is now made to the following Detailed Description, taken in conjunction with the accompanying Drawings, in which:

FIG. 1 is an illustration of a general computer architecture of a computer system within which the present invention may operate;

FIG. 2 is a graphical illustration of a sequence editor in accordance with the present invention;

FIG. 3 is an illustration of placement of a part within a channel;

FIG. 4 is an illustration of the play tool;

FIG. 5 is an illustration of placement of an audio player part;

FIG. 6 is an illustration of the play tool in conjunction with the audio player part;

FIG. 7 is an illustration of the placement of a video player part;

FIG. 8 is an illustration of placement of the play tool in conjunction with the video player part;

FIG. 9 is an illustration of placement of an image part;

FIG. 10 is an illustration of the use of a co-occur tool;

FIG. 11 is an illustration of the use of a meet tool;

FIG. 12 is an illustration of use of a hide spacer;

FIG. 13 is an illustration of placement of a time event specifier;

FIG. 14 is an illustration of use of the move spacer;

FIG. 15 is an illustration of use of the rate spacer;

FIG. 16 is a schematic illustration of a multimedia title; and

FIG. 17 is a schematic illustration of a multimedia title and use of an event specifier.

DETAILED DESCRIPTION OF THE INVENTION

Referring first to FIG. 1, there is shown a general architecture 18 of a system of the type within which the present invention operates. Architecture 18 comprises a main bus 12, which serves to interconnect the various components, including at least some of the following (and possibly other additional components): CPU 14, floating point unit 16, bus controller 28, video RAM 22, dynamic RAM 24, static RAM 26, digital signal processor 29, internal hard disk 30, external memory device 32 (connected for example via a SCSI port 34), external network devices 36 (communicating for example over an ethernet network 38, and connected via SCSI port 34), a display device 40 (such as a CRT), printer 42, keyboard 46, and pointing device 48 (such as a mouse, track ball, etc.). The interconnection and functioning of each of these components in an operating computer system are well known in the art. However, the function and significance of particular components will be discussed in further detail where appropriate.

While the discussion that follows describes the invention in terms of the software programming environment commonly known as object oriented software design (such as a ScriptX from Apple Computer), it will be appreciated that the functionality of the present invention is not limited to such environments. However, as support for one possible environment in which the program invention operates, a description of certain relevant details of object oriented programming environments is presented.

In an object oriented programming language, the fundamental data structures of the language are objects, programming elements that contain a specification of both data and actions. Data is specified in local variables, referred to as instance variables, contained within the object. Actions, or methods, are sub-programs that operate on the variables. Methods are invoked by messages sent from one object to itself or other objects.

There are two fundamental object constructs: classes and instances. Classes have a set of variables and methods and can be thought of as templates for object types. Instances (or objects) are particular run-time instantiations of a class. An object provides storage for variables defined in its class. Each instance has its own version of those variables. When a message is sent to an object, the object looks to methods defined in its class to determine how the message is to be executed.

Classes are defined in terms of a class hierarchy. A class inherits all variables and methods from classes higher in the hierarchy, referred to as super classes. The inheriting class may be referred to as a subclass or child of the superclass. The variables and methods inherited from a superclass are available to the inheriting class as if they were defined within the inheriting class. Note that if a subclass has multiple superclasses, all the variables and methods from each superclass are inherited. This is often referred to as multiple inheritance.

There are two types of variables that may be defined within a class. The first are class variables, whose storage is shared by all instances of the defining class. Class variables are typically used for coordination between classes and instances. The second are instance variables, which are variables specific to each particular instance. That is separate storage space is allocated to each instance variable of each instance.

Methods specify the action to be performed when a message is sent to an object. When the message is sent to an object, the message name is matched against method names defined for that object. The method actually executed is the method with the same name as the message that is defined lowest in that object's class hierarchy.

Messages can have parameters. When a message is sent, all actual parameters are fully evaluated before a method is located for that message. Methods have formal parameters, corresponding to the actual parameters in the message, and temporary variables. Temporary variables may be created when the message is executed, and are available only with the scope of that method while the method is executing.

Programs perform by sending messages to objects which invoke a method in an object's class or class hierarchy. Often, methods send messages to other objects, invoking other methods. Typically, each method returns a result to the sender. At the heart of many languages, and especially an object oriented language, is a program segment for interpreting and evaluating messages and keeping track of where to return when methods call other methods.

Object oriented programming environments can be implemented with many programming languages. Some common examples of object oriented environments are: the C++ environment, the Smalltalk environment, and the ScriptX environment.

Software program code which employs the present invention is typically stored in memory, such as the internal hard disk 30, from which a developer may access the code. For distribution purposes, the software program code may be embodied on any of a variety of known media for use with a data processing system, such as a diskette or CD-ROM, or may be distributed over a network of some type to other computer systems for use by users of such other systems. Such techniques and methods for embodying software code on media and/or distributing software code are well known and will not be further discussed herein.

The present invention will now be discussed in conjunction with an example and with reference to FIGS. 2–17. As previously stated, one of the most difficult problems facing creators of applications that incorporate multimedia is the

time and event synchronization problem. An example of this might be where an application requires the following types of processing: 1) the pressing of a push button causing an animation to play; 2) during the course of the animation, a piece of audio is also played; 3) at the simultaneous ending of the animation and audio, a video plays, the second half of which plays at twice the rate of the first half; 4) independent of 1–3 above, an image appears ten seconds after the push button is pressed, remains visible for 20 seconds while moving across the screen, and then disappears; and 5) shortly after the video begins an exit push button appears on the display.

Referring to FIG. **16**, a real world example that might require this type of processing could be a computer based training application (a multimedia title **300**) for a student in an automotive service center. The student would click on an arbitrary part, a push button **302**, on a computer to view information about how to assemble a previously selected automobile part. The arbitrary event, i.e., clicking the push button, would trigger a sequence of time synchronized processing **304** starting with an animation part sequence **306** that showed the sub-components of the automobile parts connecting themselves together so the student could learn how the componentry is assembled. The audio player (audio part **308**) that runs coincident to the animation would be an oral description of how the assembling process is accomplished. When the animation and audio complete, a video (video part **310**) is played that gives a 360 degree view of the assembled auto part for perspective purposes and completeness of the student's background. Due to time constraints, the second half of the video plays at twice the rate of the first half. Independent of the above processing, an image (image part **312**) is displayed 10 seconds into the playing of the animation, and moves across the screen for 20 seconds. The image is a welcome message wishing the student good luck. Also independent of the above processing, shortly after the video begins playing, an exit push button (push button part **314**) appears at the bottom of the screen allowing the student to exit the video if it is deemed unnecessary to further understanding.

To solve the synchronization problem as described above, a sequence editor in accordance with the present invention is provided. The sequence editor is a mechanism that allows a user to synchronize relative time, absolute time and event time together in an easy to use manner. The solution assumes that items for synchronization (such as an animation, video, audio, image) have been previously identified and selected by a user. There are several tools which allow selection of these types of items for synchronization and further description is not required.

The sequence editor approach to synchronization starts with events. Events are unpredictable and often occur randomly. They can be caused by user interaction or other asynchronous activity. Examples of events could be a push button being clicked, a video playing to completion, an absolute time being reached, or an error condition occurring. An event is defined by the program creator, and is a notification of some occurrence of some thing. The sequence editor organizes its time based specification by event. In other words, a user specifies an event that will initiate a sequence of time based processing. In the example that follows, the event is the clicking (selection of) a push button, and the time based processing is the animation and media related playing.

Referring now to FIG. **2**, a sequence editor screen **60** is shown. Directly beneath a title bar **62**, is a toolbar **64**. On the toolbar **64** are a plurality of convenience tools including a co-begin tool **66**, a co-end tool **68**, a co-occur tool **70**, and a meet tool **72**.

The co-begin tool **66** allows parts which are selected, as will be discussed later, to begin at exactly the same time. Similarly, the co-end tool **68** allows parts to end at exactly the same time. The co-occur tool **70** allows parts to begin and end at exactly the same times, while the meet tool **72** allows one part to begin immediately upon the ending of another. The use of these tools will be discussed in greater detail below. Beneath the toolbar **64** is a part starter **74** and an event starter **76**. By clicking on the scroll arrows **78** and **80**, respectively, drop down boxes appear for the selection of an appropriate, arbitrary part and an appropriate, arbitrary event.

Below the part and event starters **74** and **76**, there is a zoomer **82** and a time-line **84**. The zoomer **82** allows the setting of a time value for each tick interval on the time-line **84**. By moving a slider **86**, a tick interval setting may be selected. As shown in FIG. **2**, the tick interval setting is for 0.5 seconds. Therefore, each tick interval on time-line **84** represents 0.5 seconds.

Directly below the zoomer **82** are a plurality of adjustable, drag and drop tools for use by the developer. A play spacer **88** allows the developer to indicate how long a time based media part (i.e., processing) is to play. A hide spacer **90** is used to hide a part that would otherwise be visible. A move spacer **92** causes an object to move along a path while continuing to play or be visible. A rate spacer **96** changes the rate that the media is played for the length of time of the spacer.

Time event specifiers are also located beneath the zoomer **82**. A time event specifier **98** specifies an event to be generated at a particular point in time, as will be subsequently discussed in greater detail. A periodic time event specifier **94** specifies an event to be generated at a particular point in time and at subsequent intervals thereto.

A bullpen **100** is used as a staging area for parts on a scene that have been selected previously. As shown in sequence editor screen **60**, the parts for scene one are stored in the bullpen **100**.

Below the time-line **84**, are a plurality of visibility channels **102**, **104**, **106**, **108** and **110**. Corresponding to each visibility channel **102** through **110**, are thumbnail areas **112**, **114**, **116**, **118** and **120**. The thumbnail areas are used to place parts which will be synchronized by the sequence editor. The visibility channels are used to specify the period of time that the part within the appropriate thumbnail area is visible or is played.

As indicated above, the remaining FIGS. **3–17** will be used to describe a real world example of a multimedia application that an application developer might create. The example is the pressing of a push button to cause an animation to play (an animation showing sub-components of an automobile part connecting itself together). During the course of the animation, a piece of audio is also played (an oral description of how the assembling process is accomplished, plus some music). At the simultaneous ending of the animation and audio, a video plays (a 360 degree view of the assembled part for perspective purposes and completeness of the student's background). The second half of the video is to play at twice the rate of the first half. Independently, an image will appear 10 seconds after the push button is pressed, remain visible and move across the screen for 20 seconds and then disappear (a welcome message). Shortly after the video begins, an exit button appears on the display (allows the student to exit the video if deemed unnecessary).

Referring to FIG. 3, the developer has set the zoomer 82 for a two second interval. Therefore, each tick mark on the time-line 84 will indicate two seconds with each larger tick mark 120 indicating ten seconds and each largest tick mark 122 will indicate twenty seconds. The developer has also dragged the animation part 124 from the bullpen 100, and dropped it into thumbnail area 112. Upon placement of the animation part 124 in the thumbnail area 112, the visibility channel 102 (See FIG. 2) is changed to a show block 103 which indicates the animation part 124 can appear along the entire time-line 84.

Referring to FIG. 4, the play tool 88 has been dragged and dropped onto the show block 103 as indicated by a play icon 130 and a play area 131. By dragging a size handle 132 to line up under a 100 second tick mark 134 on the time-line 84, one can see that the animation player will play from the moment the push button is clicked until 100 seconds of absolute time have elapsed.

Referring to FIG. 5, an audio player part 140 has been dragged from the bullpen 100 and dropped into the thumbnail area 114. As discussed above with reference to animation player 124, upon dropping the audio player 140 into the thumbnail area 114 a visibility block 105 is placed in the visibility channel 104 (FIG. 2). The visibility block 105 extends throughout the length of the time-line 84. For the case of a non-visual time based media part (i.e., audio), visibility implies that the part can be played.

Referring to FIG. 6, the play tool 88 is dragged and dropped onto the visibility block 105 as indicated by a play icon 142 and a play area 143. Since a piece of audio plays for a finite period of time, the audio player part 140 can be seen to have an approximate duration of about 30 seconds. As will be discussed in greater detail, it is intended in this example to make the audio coincident throughout with the animation. The present invention allows such synchronization with a few simple steps.

Referring to FIG. 7, a video player part 150 has been dragged from the bullpen 100 and dropped onto the thumbnail area 116. As before, the visibility channel 106 (FIG. 2) is changed to indicate a show block 107 throughout the time-line 84. The play tool 88 is dragged and dropped onto the show block 107 (FIG. 8), as indicated by a play icon 152 and a play area 153. As with the audio portion of audio player part 140, the video player part has a finite length and therefore plays for a specified amount of time which is shown as the play area 153.

Referring to FIG. 9, the developer has selected Image 1 part 160 from the bullpen 100 and dropped it into thumbnail area 118. The visibility channel 108 is then changed to a show block 109 throughout the time-line 84.

As was stated in the real world example above, the animation and audio were to start and stop at the same time. By utilizing the present invention, the developer may simply ensure that this occurs. Referring to FIG. 10, by selecting first the animation player play area 131 and then the audio player play area 143 and then the co-occur tool 70, it is ensured that the animation player part and the audio player part will start and stop simultaneously. Since the audio was only about 30 seconds in length, the audio will obviously have to be replayed to reach the 100 second length of the animation.

Similarly, referring to FIG. 11, the video is to begin as soon as the animation ends. Thus, the developer first selects the animation player play area 131, then the video player play area 153 and, finally, the meet tool 72. By so doing, the developer has now ensured simply and quickly that the video will start upon the conclusion of the animation.

The use of the co-tools as described includes an important embodiment of this invention. These tools allow automatic implementation of time alignment of three types: 1) absolute time, 2) relative time, and 3) event driven time. In terms of background, it may be helpful to elaborate on absolute time (aT), relative time (rT), and event drive time (eT). Absolute time (aT) is time that has a fixed starting point. It may originally be specified as offset from some relative time (rT) or some event driven point in time (eT), but aT is specified overtly herein with respect to a tick mark on the time-line. Given a particular time sequence, aT is declared independent of other parts. Relative time (rT) is time that has a starting (or ending or both) point relative to some other part (or parts). Relative time (rT) is specified overtly with respect to the starting or ending of some other occurrence of something and its specific value is not known or declared. Event driven time (eT) is time that has a starting point that is defined by the occurrence of some event which may or may not have any relationship with time. An event could be a user interaction with a push button being selected or it could be an error condition. There are an infinite number of event types. Subsequent to specification of a part that is synchronized with eT, it is impossible to determine the point at which that time will occur. It will occur when that event occurs. As used herein, "initiating appropriate part behavior" means to cause the part to be, for example, displayed or played as appropriate to the type of time based media part. It is to be understood that appropriate part behavior may encompass any behavior that the developer desires.

Bringing rT back to our example, the synchronization of the audio and animation via the co-occur tool is an implementation of specifying both the starting and ending rTs of the audio with respect to the animation. The synchronization of the video and animation via the meet tool is an implementation of specifying the starting rT of the video with respect to the ending time of the animation.

Bringing aT back to our example, the synchronization of the disappearance of the image at the absolute time mark of 30 seconds on the time-line is an implementation of specifying the starting aT of the image disappearance with respect to an absolute time on the time-line. The specification is achieved by overtly dragging the hide block start time to the 30 second tick mark.

Bringing eT back to our example, the initiation of the entire sequence of time based processing starts at the unknown time of selection of the clicked event of the push button part specified by the part and event starter drop down lists.

Referring now to FIG. 12, it is still required to synchronize the Image 1 part 160. As required by the example, Image 1 part 160 is to appear ten seconds after the animation starts, remain visible for 20 seconds while moving across the screen, and then disappear. Therefore, the developer drags a hide spacer 90 and drops it onto the show block 109, as indicated by a hide icon 180 and a hide area 181. For convenience purposes only, the example shown has a hide area width of a predetermined value of ten seconds. Therefore, the hide icon 180 will cause the Image 1 part 160 to be hidden for the required ten seconds after the animation starts. The developer then grabs another hide spacer 90 and drops it onto the show block 109 as indicated by a hide icon 182. Then, by grabbing a size handle 184 and dragging to a wrap corral 186, it is ensured that the Image 1 part 160 will disappear after being visible for 20 seconds and not reappear, as shown by hide area 183. A wrap corral can be considered as infinity in the present invention. By using the wrap corral, a developer can "hook" the end time of a show or play block and thus have it continue to infinity.

In addition, continuing to use the above example, the image which is a welcome message to the student is to move in a straight line across the screen from left to right during the 20 seconds that it is visible. To accomplish this, the developer would select the move spacer **92** from the palette of tools and drop it on the visibility block **109** as indicated by icon **210** and area **211** (FIG. **14**). The visibility block **109** (FIG. **12**) is now shown as area **211** from an absolute time of 10 seconds until an absolute time of 30 seconds. Of course, this sequence is relative to the event time of the clicked event. The beginning (left edge) of the move spacer area **211** is dragged to the absolute time of 10 seconds. The ending time (right edge) of the move spacer area **211** (size handle) is dragged to the absolute time of 30 seconds. This move spacer area **211** now indicates that during the course of its time (i.e., aT 10 until aT 30) the image will not only be visible, but will move along some path. To those knowledgeable in the art of animation and computer graphics, there are many well known mechanisms to define such a path. Macromedia's Director, for example, describes in their guide at pages 113 and 114 their specification technique to allow a user of their tool to specify a path for motion of an image. There is no claim herein of novelty in motion specification techniques. As a result, we will assume that "double clicking" the move spacer icon, once it is placed on the visibility block of the image, allows the user to specify a motion path for the image that is moving across the screen. The uniqueness herein is that once a path is specified, the sequence editor of the present invention has provided an easy to use mechanism that synchronizes the movement across the screen with the length of time the move spacer occupies.

Referring to FIG. **13**, the developer drags and drops a time event specifier **98** onto the video play area **153** as indicated by a specifier icon **190**. By positioning the time event specifier icon **190** (an arbitrary unique event) properly thereon, the developer may then trigger another unique event at the desired time. This unique event could then cause another sequence to occur or, as in our real world example, it could cause some other processing to make a push button appear, which allows for termination of the video, if desired.

Referring to FIG. **17**, the event specifier of FIG. **13** is graphically illustrated in use. As previously described with respect to FIG. **16**, the multimedia title **300** is created. It is desired by the developer to generate a new arbitrary sequence **320** or a new arbitrary part **322** by a unique arbitrary event **324** (i.e., at a particular time, encountering the event specifier). Thus, as per the description above, the event specifier allows generation of the unique event.

An addition to the above example, it may be desired to have the second half of the video played at twice the normal speed. This could be necessary due to the time requirements levied on the curriculum. To accomplish this, a developer would select the rate spacer **96** (see FIG. **15**) from the palette of tools and drop it on the video play area **153** as indicated by icon **220** and area **222**. Using normal drag techniques and the size handle, the area **222** is adjusted to occupy the second half of the video play area **153**. Although it is not part of the present invention, the developer could, for example, then double click on the rate spacer icon **220** and a dialog box would appear to query the developer for the rate required. The second half of the video would then play at the new rate.

Thus, the present invention provides heretofore unavailable simplicity of use for the multimedia title developer. The developer has all the tools necessary to create a synchronized multimedia title in an easy to use visual interface type application. Events and parts can be synchronized using absolute time, relative time and/or event time which allows the developer a great deal of flexibility. The sequence editor, as taught herein, provides for the developer a view of the multimedia title from the perspective of its time structure as opposed to previously available data flow or user interface perspectives.

Although the present invention has been described with respect to a specific preferred embodiment thereof, various changes and modifications may be suggested to one skilled in the art, and it is intended that the present invention encompass such changes and modifications as fall within the scope of the appended claims.

We claim:

1. A multimedia application development sequence editor for enabling a user of said sequence editor to specify a continuous display of a part of a multimedia title, comprising:

a graphical user interface which provides an at a glance relational visual layout of a plurality of multimedia parts in terms of a time structure thereof;

at least one spacer area graphically represented on said graphical user interface, wherein each of said spacer areas is visually associated with a selected one of said parts;

means, operable by said user, for dragging a graphical representation of an edge of a selected one of said spacer areas;

means, operable by said user, for dropping said graphical representation of an edge into a wrap corral, wherein said wrap corral is a special-function area graphically depicted on said visual layout; and

means, responsive to said dragging and dropping, for causing said selected part to be appropriately displayed continuously while the multimedia title is active.

2. The multimedia application development sequence editor of claim **1**, wherein said means for dragging a graphical representation of an edge of a selected one of said spacer areas further comprises:

means, operable by said user, for dragging a wrap handle affixed to said edge.

3. A method for enabling a user of a multimedia application development sequence editor to specify a continuous display of a part of a multimedia title with said sequence editor, comprising the steps of:

providing a graphical user interface with an at a glance relational visual layout of a plurality of multimedia parts in terms of a time structure thereof;

graphically representing at least one spacer area on said graphical user interface, wherein each of said spacer areas is visually associated with a selected one of said parts;

dragging a graphical representation of an edge of a selected one of said spacer areas;

dropping said graphical representation of an edge into a wrap corral, wherein said wrap corral is a special-function area graphically depicted on said visual layout; and

responsive to said dragging and dropping, causing said selected part to be appropriately displayed continuously while the multimedia title is active.

4. The method of claim **3**, wherein said step of dragging a graphical representation of an edge of a selected one of said spacer areas further comprises:

dragging a wrap handle affixed to said edge.

5. A wrap corral for enabling a user of a multimedia application development sequence editor to specify a con-

tinuous display of a part of a multimedia title, wherein said wrap corral is a special-function area graphically depicted on an at a glance relational visual layout of a plurality of multimedia parts, said layout presented on a graphical user interface in terms of a time structure of said parts, comprising:

means, operable by said user, for dragging a graphical representation of an edge of a selected one of at least one spacer areas graphically represented on said graphical user interface, wherein each of said spacer areas is visually associated with a selected one of said parts;

means, operable by said user, for dropping said graphical representation of an edge into the wrap corral; and

means, responsive to said dragging and dropping, for causing said selected part to be appropriately displayed continuously while the multimedia title is active.

6. The wrap corral of claim **5**, wherein said means for dragging a graphical representation of an edge of a selected one of said spacer areas further comprises:

means, operable by said user, for dragging a wrap handle affixed to said edge.

* * * * *

US005999173A

United States Patent [19]

Ubillos

[11] Patent Number: 5,999,173

[45] Date of Patent: Dec. 7, 1999

[54] **METHOD AND APPARATUS FOR VIDEO EDITING WITH VIDEO CLIP REPRESENTATIONS DISPLAYED ALONG A TIME LINE**

[75] Inventor: **Randy Ubillos**, Sunnyvale, Calif.

[73] Assignee: **Adobe Systems Incorporated**, San Jose, Calif.

[21] Appl. No.: **07/863,432**

[22] Filed: **Apr. 3, 1992**

[51] **Int. Cl.**[6] **G06F 15/20**

[52] **U.S. Cl.** **345/328**

[58] **Field of Search** 395/161, 154, 395/159; 343/575, 719, 715, 718; 358/182, 183, 328, 334, 346, 951, 960, 967, 977

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,538,188	8/1985	Barker et al.	360/14.3
4,746,994	5/1988	Ettlinger	360/13
4,754,342	6/1988	Duffy	360/14.3
4,868,687	9/1989	Penn et al.	360/13
4,943,866	7/1990	Barker et al.	358/335
5,101,364	3/1992	Davenport et al.	395/152
5,109,482	4/1992	Bohrman	395/154

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0 113 993	7/1984	European Pat. Off.	G11B 27/02
0 390 048	10/1990	European Pat. Off.	G11B 27/00
0 438 299 A2	7/1991	European Pat. Off.	G11B 27/034
0 469 850 A2	2/1992	European Pat. Off.	G11B 27/034
0 526 064 A2	2/1993	European Pat. Off.	G11B 27/028
2 630 572	10/1992	France	G11B 27/028
91/10321	7/1991	WIPO	G11B 27/034

OTHER PUBLICATIONS

Tonomura et al., "Content Oriented Visual Interface Using Video Icons for Visual Database Systems", IEEE Workshop on Visual Languages, pp. 68–73, Oct. 1989.

Aker, Sharon Z., "Roll Over, Walt Disney," MacUser, pp. 150–152, 154, 156–157, Mar. 1988.

Clouser, Charles, "MediaTracks 1.0." MacWorld, pp. 198 and 200, Jan. 1991.

Biedny, David, "MacroMind Director 2.0," MacWorld, pp. 249 and 251, Oct. 1990.

Krutz, Jamie, "KNVB's Macs go video.", MacWeek, V2 N43 pS84(3), Dec. 1988.

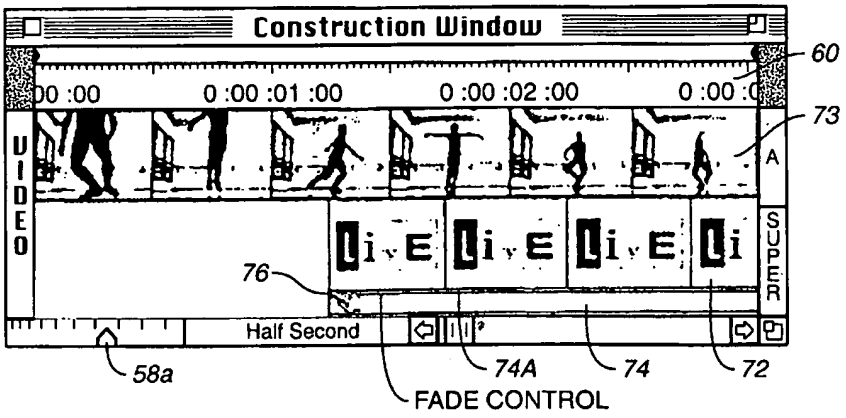
(List continued on next page.)

Primary Examiner—Robert W. Beausoliel, Jr.
Assistant Examiner—Stephen Elmore
Attorney, Agent, or Firm—Fish & Richardson P.C.

[57] **ABSTRACT**

A method and apparatus, in which video clips (and optionally also still image and audio clips) are stored as digital data in a computer memory, selected clips are displayed in elongated tracks on a display screen, and editing operations are performed on the clips in response to manipulation of displayed cursors and icons to assemble and preview an edited video program. Preferably, the inventive apparatus is a computer system programmed to display video, still image, and audio clips, and special effect icons, in tracks along a displayed time line. The system assembles a video program from stored clips in response to arrangement of displayed clips and special effect icons in a desired sequence along the time line. The computer system is preferably programmed: to select a new in or out point for a clip by positioning a cursor at an edge of the displayed clip and dragging the edge relative to the time line, to select a special effect transition between displayed clips by positioning a transition icon in a special track in alignment with overlapping portions of the clips, to select special effect parameters by manipulating an icon in a special effects track, to filter selected video clips with a mosaic filter having user-selectable time-varying filter characteristics, and to control superimposition of an overlay clip with a main clip in response to manipulation of a level control icon displayed in alignment with the overlay clip.

36 Claims, 7 Drawing Sheets



U.S. PATENT DOCUMENTS

5,119,474	6/1992	Beitel et al.	345/302
5,148,154	9/1992	MacKay et al.	340/712
5,191,645	3/1993	Carlucci et al.	395/159
5,202,961	4/1993	Mills et al.	345/328
5,206,929	4/1993	Langford et al.	395/159
5,218,672	6/1993	Morgan et al.	395/162
5,237,648	8/1993	Mills et al.	395/133
5,274,758	12/1993	Beitel et al.	345/302
5,307,456	4/1994	MacKay	395/154
5,317,732	5/1994	Gerlach, Jr. et al.	395/600
5,353,391	10/1994	Cohen et al.	395/125
5,359,712	10/1994	Cohen et al.	395/161
5,388,197	2/1995	Rayner	395/154
5,519,828	5/1996	Rayner	395/161

OTHER PUBLICATIONS

Fraser et al., The nuts and bolts of Video F/X. (Digital F/X Inc.'s desktop video–editing system), MacWeek, V4 N35 p36(3), Oct. 1990.

"MediaMaker Makes Mac Move", Murie, Michael D., MacWeek, Mar. 26 1991, V5, n12, p. 53.

"QuickTime to Bring Dynamic Dimension to Macs", Karraker, Roger, MacWeek, Aug. 6 1991, V5 n27, p. 50.

"Macromind's MediaMaker to Make Expo Debut", Guglielmo, Connie, MacWeek, Jul. 31, 1990, V4 n26, p. 14.

"Will Picons, Micons or Dicons Supercede Icons?," Rosenthal, Steve, MacWeek, Sep. 26, 1989, V3 n34, p. 32.

Manual entitled "Adobe Premiere™ User Guide–Macintosh Version," Sep. 1991 (94 pages).

Krieg, "Multimedia Computers and the Future of Film/Video Editing," Fernseh–und Kino–Technik, vol. 45, No. 5, pp. 252–254; 256 and 258 (1991) (with English translation).

Klingler et al., "Fusion of Digital Image Processing and Video on the Desktop," pp. 64–66, Digest of Papers, COMPCON, 37th IEEE Computer Society Int. Conference (Feb. 1992).

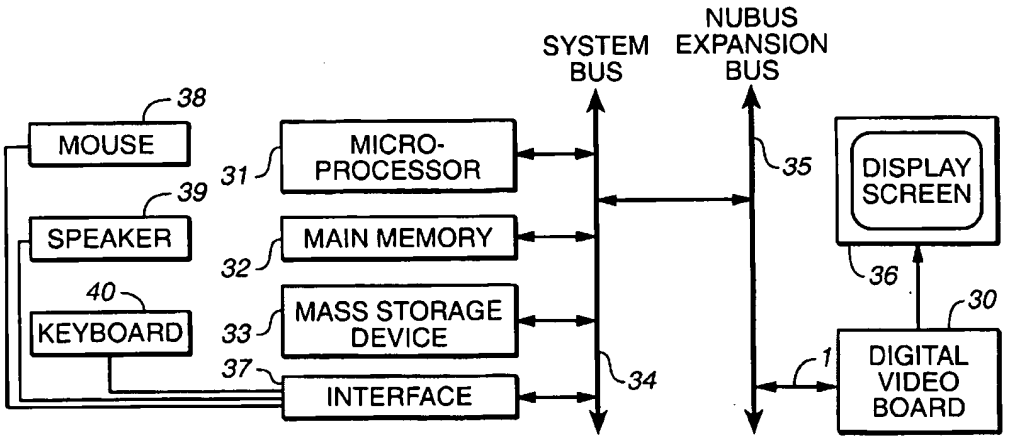

SYSTEM BUS
NUBUS EXPANSION BUS
38
MOUSE
39
SPEAKER
40
KEYBOARD
31
MICRO-PROCESSOR
32
MAIN MEMORY
33
MASS STORAGE DEVICE
37
INTERFACE
34
35
DISPLAY SCREEN
36
30
1
DIGITAL VIDEO BOARD

FIG._1

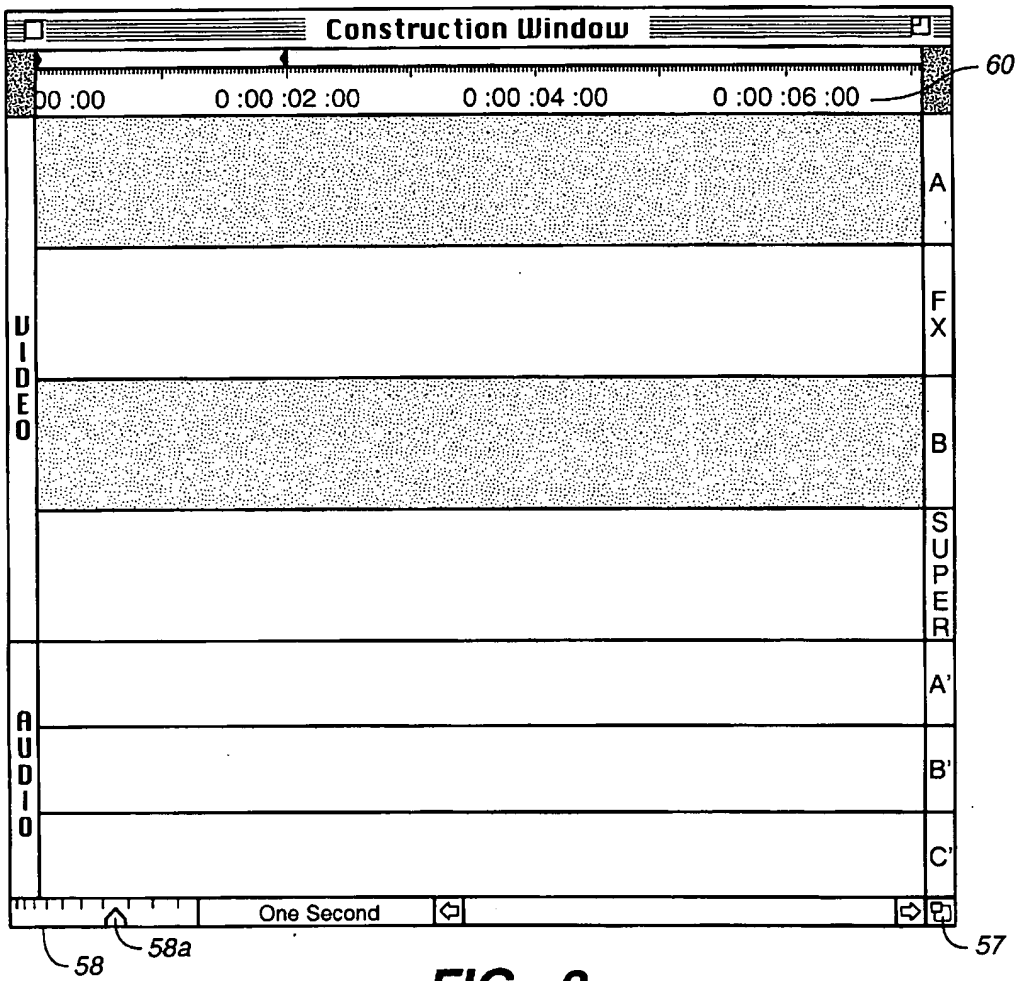

Construction Window
60
00 :00
0 :00 :02 :00
0 :00 :04 :00
0 :00 :06 :00
VIDEO
A
FX
B
SUPER
AUDIO
A'
B'
C'
One Second
58
58a
57

FIG._2

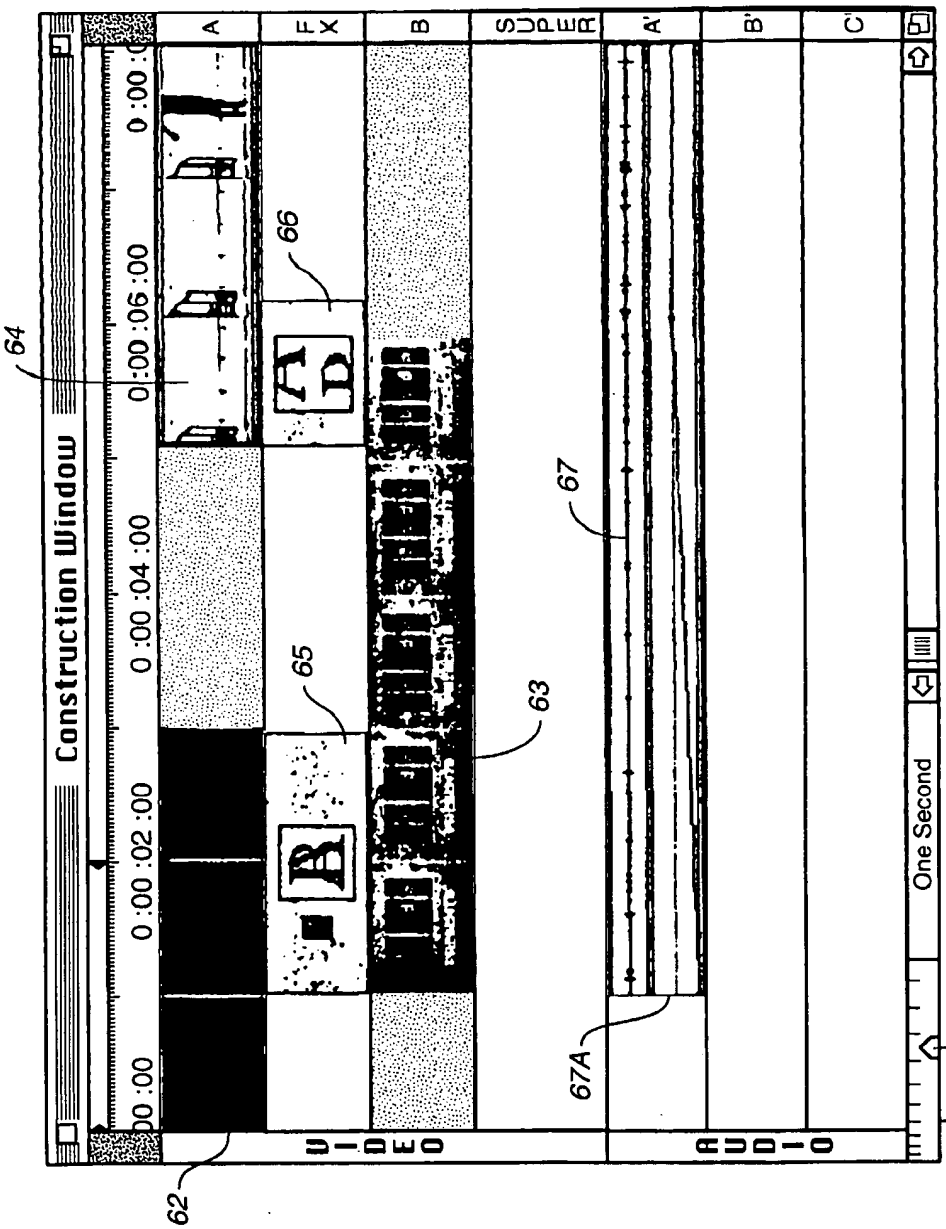

Construction Window
A
FX
B
SUPER
A'
B'
C'
0:00
0:00:02:00
0:00:04:00
0:00:06:00
VIDEO
AUDIO
One Second
62
63
64
65
66
67
67A
58
58a

FIG._3

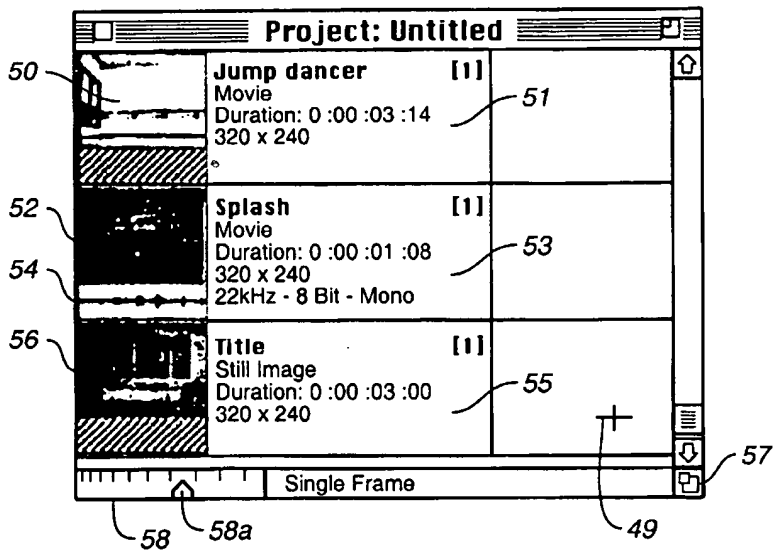

Project: Untitled
Jump dancer [1]
Movie
Duration: 0 :00 :03 :14
320 x 240
Splash [1]
Movie
Duration: 0 :00 :01 :08
320 x 240
22kHz - 8 Bit - Mono
Title [1]
Still Image
Duration: 0 :00 :03 :00
320 x 240
Single Frame
50
51
52
53
54
55
56
57
58
58a
49

FIG._4

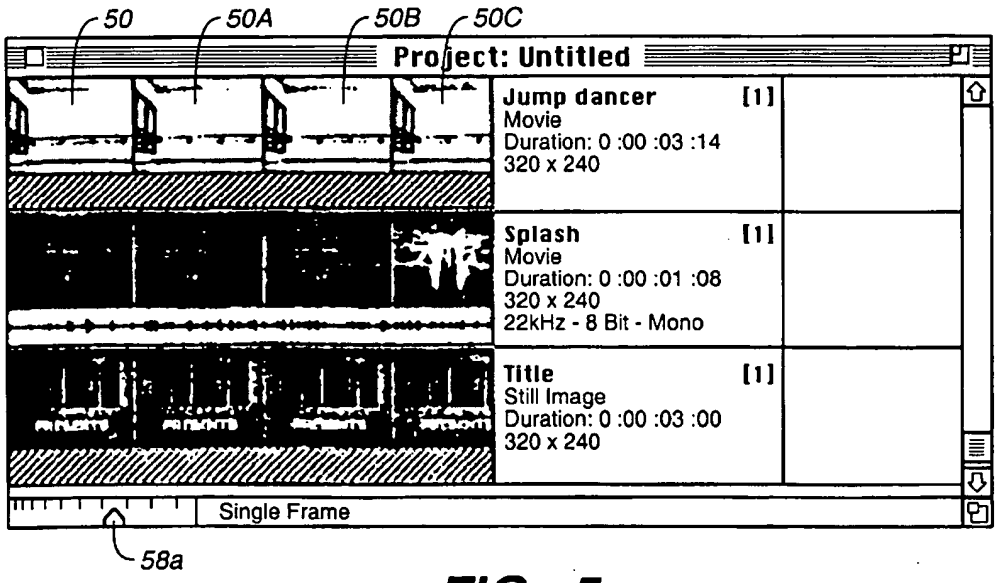

50
50A
50B
50C
Project: Untitled
Jump dancer [1]
Movie
Duration: 0 :00 :03 :14
320 x 240
Splash [1]
Movie
Duration: 0 :00 :01 :08
320 x 240
22kHz - 8 Bit - Mono
Title [1]
Still Image
Duration: 0 :00 :03 :00
320 x 240
Single Frame
58a

FIG._5

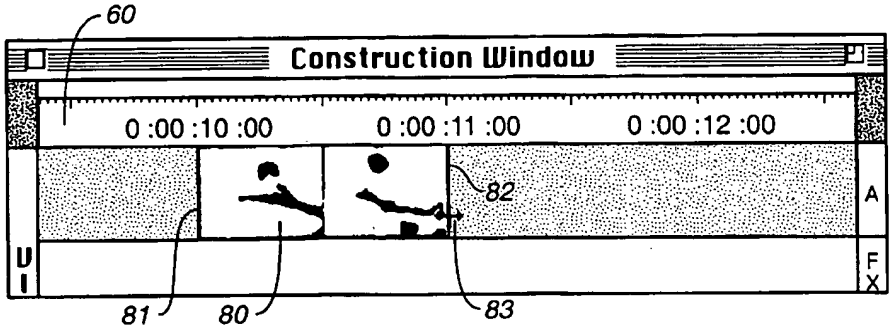


FIG._6

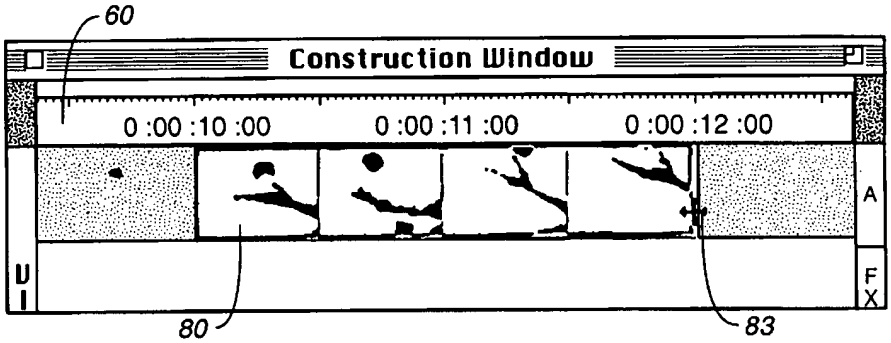


FIG._7

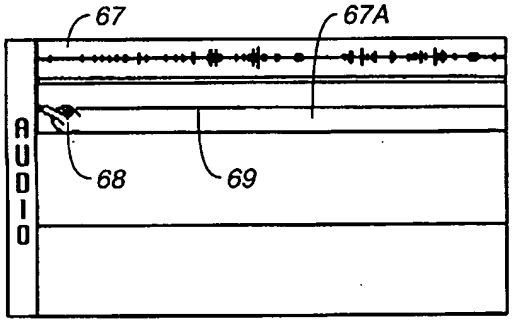


FIG._8

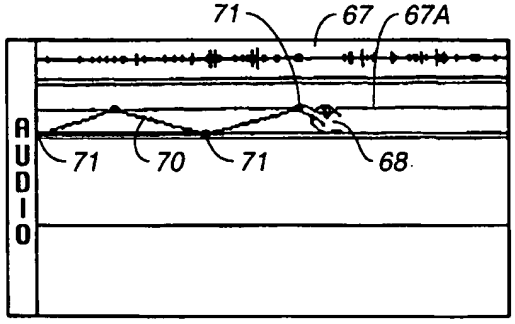


FIG._9

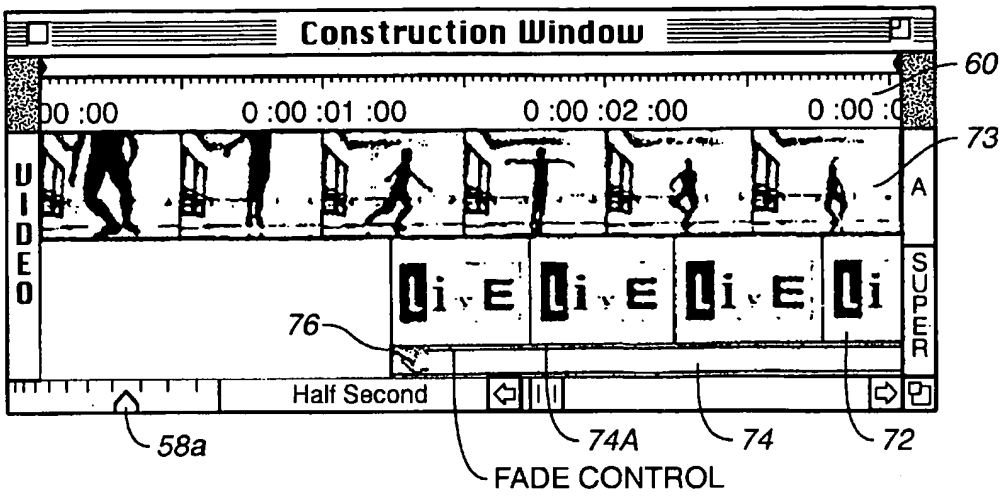

Construction Window
60
00 :00
0 :00 :01 :00
0 :00 :02 :00
0 :00 :0
73
VIDEO
A
SUPER
76
Half Second
58a
74A
74
72
FADE CONTROL

FIG._10

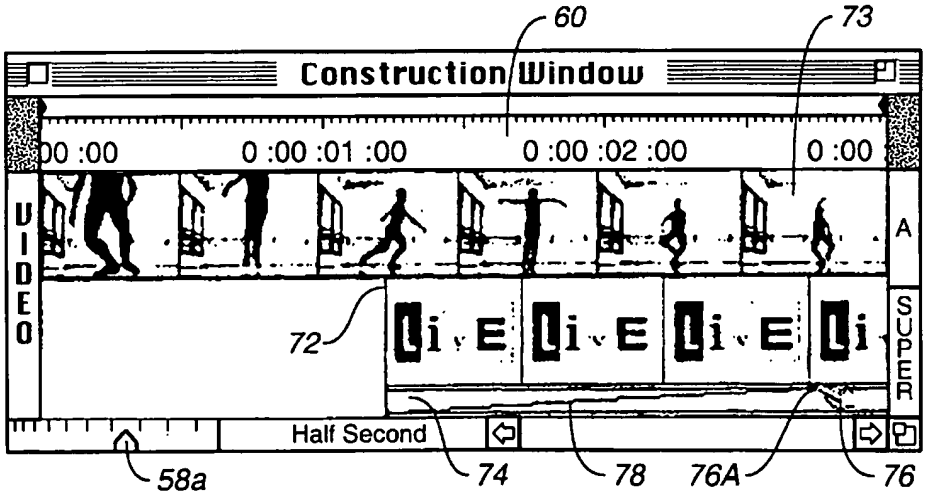

60
73
Construction Window
00 :00
0 :00 :01 :00
0 :00 :02 :00
0 :00
VIDEO
A
SUPER
72
Half Second
58a
74
78
76A
76

FIG._11

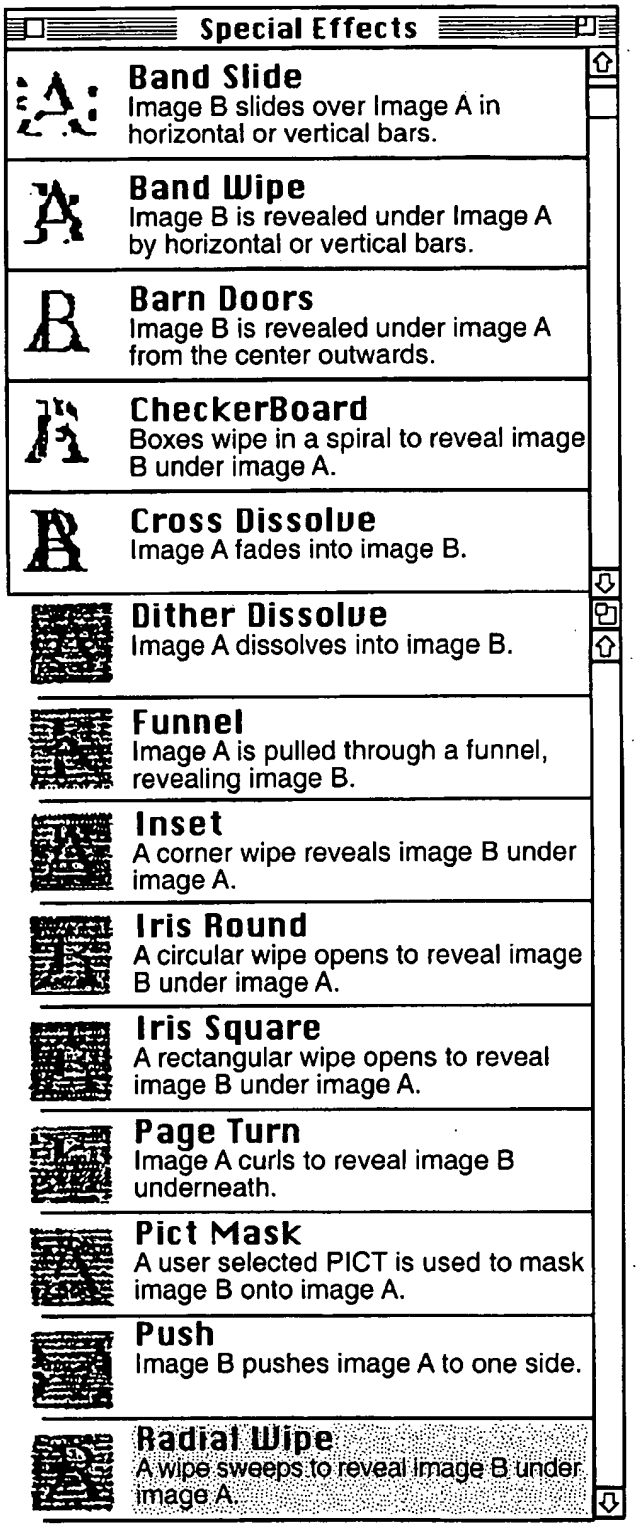

Special Effects
Band Slide
Image B slides over Image A in horizontal or vertical bars.
Band Wipe
Image B is revealed under Image A by horizontal or vertical bars.
Barn Doors
Image B is revealed under image A from the center outwards.
CheckerBoard
Boxes wipe in a spiral to reveal image B under image A.
Cross Dissolve
Image A fades into image B.
Dither Dissolve
Image A dissolves into image B.
Funnel
Image A is pulled through a funnel, revealing image B.
Inset
A corner wipe reveals image B under image A.
Iris Round
A circular wipe opens to reveal image B under image A.
Iris Square
A rectangular wipe opens to reveal image B under image A.
Page Turn
Image A curls to reveal image B underneath.
Pict Mask
A user selected PICT is used to mask image B onto image A.
Push
Image B pushes image A to one side.
Radial Wipe
A wipe sweeps to reveal image B under image A.

FIG._12

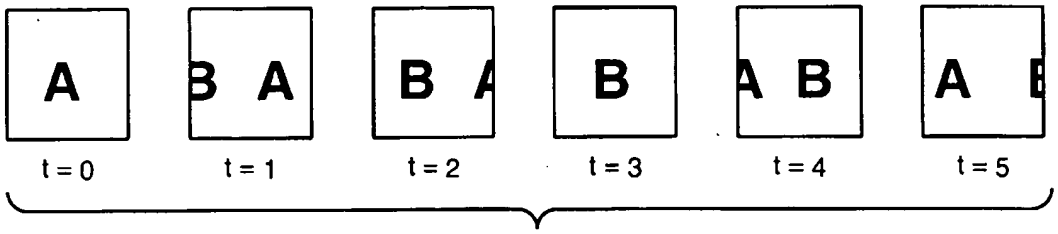


FIG._13

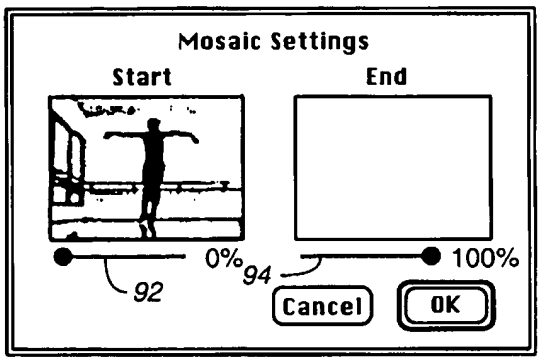


FIG._14

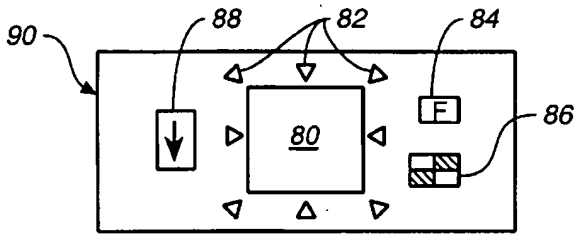


FIG._15

METHOD AND APPARATUS FOR VIDEO EDITING WITH VIDEO CLIP REPRESENTATIONS DISPLAYED ALONG A TIME LINE

FIELD OF THE INVENTION

The invention is a computer-implemented video editing method and apparatus in which stored video clips are displayed, and an edited video program is assembled from selected ones of the clips. The apparatus of the invention is a computer system programmed to perform video editing operations in response to manipulation of displayed menus and icons.

BACKGROUND OF THE INVENTION

Throughout this specification, the term "video clip" will denote one or more consecutive frames of video data, the term "still image clip" will denote data representing one or more identical frames of still image data (or text), and the term "audio clip" will denote audio soundtrack data corresponding to one or more video clips. A video clip (or a still image clip) can be a scene of an edited video program.

Throughout this specification, the expression "display of a clip" and denotes display (at a single time) of a representation of the data defining the clip, unless qualified to read "sequential display of a clip". Similarly, "displayed clip" denotes a simultaneously displayed representation of the data defining the clip, unless qualified to read "sequentially displayed clip". The expressions "sequential display" of a program, "sequential display" of a clip, and variations on these expressions, denote the non-simultaneous (time-sequential) display of a sequence of representations of data sets, where the data sets collectively define a program or clip.

Throughout this specification, including in the claims, the expressions "edited video program," "video program," and "program" are used interchangeably to denote a sequence of video clips or still image clips (or video clips and still image clips), a transition between each pair of consecutive video and/or still image clips, and optionally also a sequence of audio clips. Each of the transitions between video (and/or still image) clips can be a simple "cut," in which the last frame of a first clip is concatenated with the first frame of a second clip. Alternatively, one or more of the transitions can be a more complicated (special effect) transition, such as a wipe, fade, or dissolve. For example, an edited program can consist of a first video clip, followed by a simple cut to a second video clip, followed by a dissolve transition to a third video clip.

Typically, an edited video program consists of a sequence of video and still image clips (with special effect transitions between consecutive pairs of these clips), and two or more sequences of audio clips (for example, two sequences of audio clips representing left and right channels of an audio soundtrack).

It is conventional to define an edited video program by employing a programmed computer to perform video editing operations in which the computer processes stored digital data representing segments of the video program. A conventional system of this type is disclosed in U.S. Pat. No. 4,538,188, issued Aug. 27, 1990, to Barker, et al. The system of U.S. Pat. No. 4,538,188 employs a computer to control the display and manipulation of pictorial labels (or label pairs), each representing an unedited video segment (or transition between consecutive video segments), in order to assemble an edited video program from unedited video segments. The system also assembles (from the unedited video segments) and displays a sequence of the pictorial labels, which represent scenes (and transitions between scenes) of the edited program. The video segments are stored on video tape, as is the assembled video program. The system has controls which enable a user to specify a variety of transitions between consecutive video segments.

Other video editing systems have been proposed which include means for storing unedited video clips (on video tape or laser video disks), and computer means for controlling the display of selected unedited video clips (or frames of video clips) and generating an edit list in response to user-entered commands.

For example, U.S. Pat. 4,746,994 (issued May 24, 1988, to Ettlinger) discloses a computer-based video editing system in which unedited clips are stored on video tape recorders or video disk players. A computer system enables the user to control the video tape recorders and generate an edit list. The computer system displays a sequence of menus which prompt the user to perform various editing operations (such as displaying desired frames of the unedited clips, shuttling frame-by-frame through a stored unedited clip, adding edits to an edit list, and playing back the sequence of clips defined by the edit list). The user may select various ones of the editing operations by actuating a light pen.

For another example, U.S. Pat. 4,754,342 (issued Jun. 28, 1988, to Duffy) discloses a computer-based video editing system in which unedited clips are stored on video disk players. After generating an edit list, the user may command the system to splice clips in accordance with the edit list and play back an edited show. By using a control console with control buttons and a control dial, the user may command the system to display individual frames of the stored clips, or "roll" one or more clips (or an edited sequence of clips) in forward or reverse motion, at any of a variety of speeds.

SUMMARY OF THE INVENTION

The invention is a method and apparatus for video editing, in which video clips (and optionally also still image clips and audio clips) are stored as digital data in a computer memory, selected clips are displayed in elongated windows (known as "tracks") on a display screen, and editing operations are performed on the clips in response to manipulation of displayed cursors and icons to assemble and preview an edited video program.

The preferred embodiment of the inventive apparatus is a computer system programmed to display representations of video, still image, and audio clips at desired positions along a displayed time ruler, in tracks of a construction window. The system dynamically generates each video clip to be displayed by retrieving from storage all frames of the video clip (or every "Nth" frame of the stored clip in accordance with a user-selected time compression factor), and displaying the retrieved frames.

Animated "special effect" icons, each representing a special effect, are displayed in a separate special effects track also oriented parallel to the time ruler. Each special effect icon can represent a special effect transition between two clips (such as a dissolve, fade, and wipe).

The user instructs the computer to assemble a video program from the stored video clips (and optionally also stored audio clips and clips representing still image images or text) by arranging displayed clips and displayed special effect icons in a desired sequence along the time ruler.

In a preferred embodiment, the computer system of the invention is programmed to perform the following operations:

1. select a new "in" point, "out" point, or both, for a displayed clip (in the edited video program) in response to the positioning of a cursor at an edge of the displayed clip and manipulation of an input device to "drag" the edge along the time ruler;

2. select a special effect transition between a pair of video clips displayed in first and second parallel tracks (and having overlapping portions along the time ruler), by displaying a selected transition icon in a third parallel track in alignment with the overlapping portions of the video clips;

3. select special effect transition parameters (such as transition duration) by displaying a transition icon in a special effects track (where the special effects track is displayed below a first video track and above a second video track, and where overlapping video clips are displayed in the first and second video tracks), and positioning a cursor over the transition icon while manipulating an input device;

4. control superimposition of an overlay clip (representing video, still image, or text) with a main (video or still image) clip by displaying the clips in separate tracks, displaying a level control icon in alignment with the overlay clip, and manipulating the level control icon to display a graph representing a time-varying weighting function for combining the main and overlay clips;

5. preview a video program defined by clips displayed in a desired sequence along a time ruler of a construction window by retrieving the clips from memory, processing the retrieved clips (for example, in accordance with special effects transitions defined by transition icons displayed along the time ruler), and sequentially displaying the processed clips as a preview in a separate video window;

6. cache (in random access cache memory) all frames retrieved in response to a command to update a displayed window or to preview a program defined by clips displayed in a desired sequence along a time ruler of a construction window;

7. process one or more selected video clips by filtering the corresponding digital video data with a temporally varying mosaic filter (having user-selected time-varying filter characteristics); and

8. perform special effects processing by filling the alpha channel of a first clip with a first value (i.e., a hexadecimal "00"), filling the alpha channel of a second clip with a second value (i.e., a hexadecimal "1F"), performing a first special effect process in which selected pixels from the first clip are combined with selected pixels of the second clip to generate a processed clip, and processing alpha channel data of the processed clip to identify one or more edges of the processed clip (for example, to facilitate further special effect processing of the processed clip).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a preferred embodiment of the inventive apparatus.

FIG. **2** is a construction window displayed by a preferred embodiment of the inventive apparatus.

FIG. **3** is a construction window (of the FIG. **2** type) with two video clips, one still image clip, one audio clip, and two special effect transition icons displayed in separate tracks thereof.

FIG. **4** is a project window displayed by a preferred embodiment of the inventive apparatus.

FIG. **5** is another project window displayed by a preferred embodiment of the inventive apparatus.

FIG. **6** is a portion of a construction window having a single video clip displayed in one video track, generated in an embodiment of the invention.

FIG. **7** is a modified version of the FIG. **6** display, which is produced by dragging the right edge of the video clip of FIG. **6**, in an embodiment of the invention.

FIG. **8** is a portion of a construction window having a single audio clip displayed in one audio track, generated in an embodiment of the invention.

FIG. **9** is a modified version of the FIG. **8** display, which is produced by altering the level control icon of the FIG. **8** display, in an embodiment of the invention.

FIG. **10** is a construction window displayed by a preferred embodiment of the inventive apparatus, with a video clip and an overlay clip displayed in separate tracks.

FIG. **11** is a modified version of the FIG. **10** display, which is produced by altering the level control icon of the FIG. **10** display.

FIG. **12** is a special effects menu displayed by a preferred embodiment of the inventive apparatus, including a set of animated special effect transition icons.

FIG. **13** is a set of six transition icon displays, representing the animated display of a special effect transition icon, generated in an embodiment of the invention.

FIG. **14** is a mosaic filter control menu displayed by a preferred embodiment of the invention.

FIG. **15** is a special effect transition icon of a type displayed by a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference to FIG. **1**, a preferred embodiment of the invention includes programmed microprocessor **31**, main memory **32** (a random access memory), mass storage device **33**, and interface **37**, all connected along a system bus **34**, and mouse **38**, audio speaker **39**, and computer keyboard **40** connected to interface **37**.

Expansion bus **35** is connected to system bus **34**, and digital video board **30** and computer monitor **36** (having a display screen) are connected along expansion bus **35**. Microprocessor **31** runs the operating system and applications software that controls the operation of digital video board **30** (and the other system components). Preferably, this software includes Apple QuickTime system extension software available from Apple Computer, Inc. (or software equivalent thereto) for integrating video and audio data into Macintosh applications.

Appropriately programmed microprocessor **31** performs all necessary digital processing operations (including special effects processing and filtering, including mosaic filtering to be described below) on data sets received from memory **32** or **33**. Board **30** drives monitor **36**, to display (in windows on the monitor's screen) data transferred to bus **35** from system bus **34** after undergoing special effects processing, filtering, or other processing in microprocessor **31**. Preferably, board **30** processes 24-bit color video data for display on monitor **36**, and microprocessor **31** has 32-bit addressing capability identical or equivalent to that of the Apple System 7 product available from Apple Computer, Inc.

Main memory **32** stores the program and data sets for microprocessor **31**, and preferably has at least eight megabyte capacity. Microprocessor **31** and main memory **32** are preferably implemented as a Macintosh™ II (or later model) computer having at least eight megabytes of random-access memory.

Mass storage device **33** is typically a hard disk (preferably having at least 80 megabyte capacity), but can alternatively

be a magneto-optical disk or other read/write device, or (in some embodiments) a read-only device such as a CD ROM.

Microprocessor **31** and mass storage device **33** communicate with each other (and with main memory **32**) over system bus **34**. Bus **34** also allows data communication to expansion bus **35**. In the Apple Macintosh™ II family of personal computers, expansion bus **35** adheres to the NuBus standard. Additional information on NuBus operation is available from Texas Instruments, Dallas, Tex.

Additional details regarding preferred embodiments of hardware (and software) for implementing the invention are set forth in the "Adobe Premiere™ User Guide- Macintosh™ Version," dated September 1991, available from Adobe Systems Incorporated, which document is incorporated herein by reference.

Microprocessor **31** is programmed with editing software and user interface software (to be described below). The user interface software accesses the editing software (and the other system software) in response to commands entered by the user using mouse **38** or keyboard **40**, and is capable of instructing board **30** to generate displays of the type shown in FIGS. **2–14** on computer monitor **36**.

Digital data representing video clips, still image clips, and audio clips are stored in mass storage device **33**. Microprocessor **31** "caches" selected portions of this data, including frames of data selected for display on monitor **36**, by copying the data into random access memory **32** (which will sometimes be denoted herein as a "cache" memory).

When a user initiates an editing project, the system opens, and displays on monitor **36**, a project window (such as that shown in FIG. **4** or FIG. **5**) and a construction window (such as that shown in FIG. **2**). By manipulating keyboard **40** and/or mouse **38**, the user selects stored video clips (and optionally also still image clips and audio clips) from mass storage device **33**. In response to selection of each clip, programmed microprocessor **31** generates an image ("thumbnail") for the clip, and displays the thumbnail (and corresponding identifying text) in the project window. FIG. **4** shows four such thumbnails **50**, **52**, **54**, and **56** displayed in a project window. FIG. **4** also shows identifying text **51** for a video clip having thumbnail **50**, identifying text **53** for a video clip having thumbnail **52** (and a corresponding audio clip having thumbnail **54**), and identifying text **55** for a still image clip having thumbnail **56**. The identifying text preferably includes the size in pixels (i.e., 320 pixels×240 pixels) of each video or still image clip, and the clip duration (in the SMPTE format "hours:minutes:seconds:frames").

The thumbnail for a video clip is (in a preferred embodiment) one (possibly reduced) displayed frame thereof, such as thumbnail **50** and **52** in FIG. **4**. For a still image clip, the thumbnail is (in a preferred embodiment) the corresponding image (or text), such as thumbnail **56** in FIG. **4**. For an audio clip, the thumbnail is (in a preferred embodiment) a graph of a corresponding audio signal waveform, such as thumbnail **56** in FIG. **4**.

Preferably, the system is programmed to enable a user to enlarge the project window by manipulating mouse **38** to position cursor **49** on size icon **57**, and then dragging size icon **57** toward the right (on the screen of monitor **36**). In response to enlargement of the project window, the system will display thumbnails representing additional frames of each previously selected video clip (each of which already has a thumbnail displayed in the project window). For example, FIG. **5** shows an enlarged version of the FIG. **4** project window, with three additional frames (**50A**, **50B**, and **50C**) of the video clip having thumbnail **50**. In response to enlargement of the project window, the system will display, side-by-side, additional copies of the thumbnail of each still image clip. For example, in the enlarged project window of FIG. **5**, four copies of still image clip thumbnail **56** are displayed.

Preferably, the user can control the time scale on which clips are displayed in the project window (i.e., the degree of time compression with which a video clip is displayed in the project window). This can be done by positioning cursor **49** on a "time unit" icon **58***a* in time compression window **58** and dragging icon **58***a* to a different station within window **58** (by manipulating mouse **38**). Preferably, icon **58***a* can be positioned at any of several stations, each representing a different time scale (and time compression factor). Examples of such stations include: "single frame" (for displaying each frame of a video clip, i.e., one frame for each 1/30 second segment of a video clip representing video data having a frame rate of 30 per second), "two frames" (one displayed frame for each two frames of a video clip), "one second" (one displayed frame for each segment of a video clip having 1 second duration), and "two minutes" (one displayed frame for each segment of a video clip having 2 minute duration).

In general, the system responds to any command requiring display of a video clip in a window (which can be a preview window, or a construction window of the type to be described below) by retrieving from storage, caching, and displaying every "Nth" frame of the selected video clip. The factor "N" is determined by the currently selected time compression factor. For example, if the user has selected the above-described "one second" time compression factor, the factor N equals 30, and the system will display every 30th frame of a selected video clip (if the clip has a frame rate of 30 frames per second).

For another example, in response to selection of a stored video clip from mass storage device **33** for inclusion in a project (and display in the project window) with time unit icon **58***a* at a "single frame" station in time compression window **58**, the system will display in the project window as many frames of the clip as will fit in the currently configured project window, and will cache all the displayed frames in cache memory **32**. If the user then enlarges the project window, the system will retrieve a larger subset of the clip's frames from cache memory **32** (and mass storage memory **33**), display this larger subset of frames in the enlarged project window, and cache the displayed frames in cache memory **32**.

For another example, in response to selection of a stored video clip from mass storage device **33** for inclusion in a project with time unit icon **58***a* at a "two frames" station in window **58**, the system will display in the project window as many frames of the time-compressed clip as will fit in the currently configured project window (i.e., only the first, third, and fifth frames, if only three frames will fit in the project window), and will cache the displayed frames in cache memory **32**. If the user then enlarges the project window, the system will retrieve a larger subset of the time-compressed clip's frames from cache memory **32** (and mass storage memory **33**), display this larger subset of frames in the enlarged project window, and cache the displayed frames in cache memory **32**.

To generate an edited video program, the user drags thumbnails of selected clips from the project window to a construction window of the type shown in FIG. **2** (by manipulating mouse **38**), and moves the thumbnails into a desired arrangement in the construction window. The con-

struction window displays a time ruler **60** in a timeline window along its top edge, and elongated windows ("tracks") oriented parallel to time ruler **60**. FIG. **2** shows seven tracks, labeled "A," "Fx," "B," "Super," "A'," "B'," and "C'," respectively.

Because the thumbnails displayed in tracks of the construction window represent entire video clips or time-compressed entire video clips, the following description of the construction window (with reference to FIGS. **2–3** and **6–11**) will refer to thumbnails displayed in tracks of the construction window as "clips."

Time ruler **60** represents the timeline of the edited video program. The user can expand the construction window by dragging size icon **57** (in the same manner described above with reference to identical icon **57** of the project window). Because a complete edited program may have duration longer than can be represented by clips displayed on a single monitor screen, the system permits the user to display any selected portion of program in the construction window. For example, the user can instruct the system to display the clips for a first portion of the program in a construction window (with the construction window displaying a first portion of time ruler **60** extending from the program start time 0:00:00 to intermediate time M:MM:MM), and thereafter instruct the system to display the clips for the remaining portion of the program in the construction window (with the construction window displaying the remaining portion of time ruler **60** extending from time M:MM:MM to a program end time N:NN:NN).

When clips have been arranged in the construction window along time rule **60**, the user can enter a "preview" command to microprocessor **31** to cause the system to generate a video program defined by the displayed clips. The manner in which the system responds to a "preview command" will be more fully described below, but can be summarized as follows: the system retrieves from mass storage memory **33** (or cache memory **32**) all data corresponding to the displayed clips, stores the retrieved frames in cache memory **32**, and displays the cached frames sequentially (as full-motion video) in a window on the screen of monitor **36** (and optionally also plays the corresponding audio soundtrack channels on speakers).

In the preferred embodiment shown in FIG. **2**, the construction window has seven tracks, all oriented parallel to time ruler **60**. Only audio clips can be displayed in the bottom three tracks (identified as tracks A', B', and C'), and only video or still image clips can be displayed in the top four tracks (identified as tracks A, FX, B, and "Super").

The large tick marks of time ruler **60** represent the time scale for the video program (with time displayed in SMPTE format), and the small tick marks represent frames or seconds (depending on the currently selected time scale). The user selects a time scale by manipulating icon **58***a* in window **58** (shown in FIG. **2**) in the same manner as described above with reference to identical icon **58***a* and window **58** of the project window of FIG. **4**. As described above with reference to FIG. **4**, icon **58***a* preferably can be positioned at any of several stations, each representing a different time scale (including, for example, "single frame," "two frames," "one second," and "two minutes" stations). For example, if the user drags a video clip into the construction window with time unit icon **58***a* positioned at a "one second" station in window **58** (as shown in FIG. **3**), the system will cache (in cache memory **32**) a time-compressed version of the video clip (i.e., it will cache one frame of the clip for each one-second segment of the clip). The system will also display all frames of the time-compressed clip in one track of the construction window (with one displayed frame per "one second" interval along time ruler **60**), and will display a version of time ruler **60** scaled for the selected "one second" time scale (having small tick marks representing frames between each pair of large tick marks, with the distance between each pair of large tick marks representing one second).

A very simple edited video program is defined by video (and/or still image) clips arranged side-by-side in a single track (i.e., track A), with the "Out" point (right edge) of each clip butting against the "In" point (left edge) of the next clip, and with no audio clips displayed in audio tracks A', B', and C'. In response to entry of a "preview" command with this arrangement of clips displayed in the construction window, the system previews the program by sequentially displaying (in a preview window) the clips that are simultaneously displayed in the construction window, in a time order corresponding to the spatial position of the clips along time ruler **60**.

To generate more complicated video programs (with overlapping clips), the user displays pairs of overlapping clips in different tracks of the construction window (as shown in FIG. **3**).

With reference to FIG. **3**, the user can specify a special effects transition between video clip **62** and still image clip **63** by positioning clip **62** in track A and clip **63** in track B, and positioning a special effect transition icon **65** in a track FX. The user adjusts "In" and "Out" points of icon **65** (the left and right edges of icon **65**, respectively) so that icon **65** coincides with the overlapping portions of clips **62** and **63** along time ruler **60**. Because icon **65** represents a cross-dissolve transition (to be described below), the FIG. **3** display specifies a video program including a cross dissolve transition from clip **62** to clip **63**, beginning at the "In" point of icon **65** and ending at the "Out" point of icon **65**.

FIG. **3** also shows a "push" special effect transition icon **66** in track FX, in alignment with overlapping portions of video clip **64** (in track A) and still image clip **63** (in track B). Because icon **66** represents a push transition (to be described below), the FIG. **3** display specifies a video program including a "push" transition from clip **63** to clip **64**, beginning at the "In" point of icon **66** and ending at the "Out" point of icon **66**.

FIG. **3** also shows audio clip **67** in audio track A', which represents a single-channel (monaural) audio soundtrack. Because the "In" point (left edge) of audio clip **67** is aligned with the "In" point of clip **63**, the soundtrack for the program will commence simultaneously with the start of the cross-dissolve from clip **62** to clip **63**, and the soundtrack will continue during display of clips **63** and **64**. Optionally, the user could add second and third audio clips to tracks B' and C' to specify a three-channel audio soundtrack (which might correspond to left and right stereo channels, and a "surround" or "sub-woofer" channel).

The system is programmed so that manipulation of audio level control icon **67A** (displayed with clip **67** in track A') allows the user to adjust the level of the corresponding audio track, in the same way that sound is mixed in record and television production. With reference to FIGS. **8** and **9**, the user positions a pointer (cursor) **68** on middle line **69** of icon **67A** (by manipulating mouse **38**). The user then "clicks" mouse **38** (activates a control on mouse **38**) to create a handle **71** (a black dot), and drags **71** handle up or down relative to line **69** to program the system to execute an audio clip fade in or fade out, respectively, and to deform a

displayed "rubber-band" level control line **70**. The default position of rubber-band line **70** is a position coincident with middle line **69**, which represents a mid-range audio level for clip **67**. Line **70** has "rubber-band" characteristics in the sense that, in response to a user's dragging of a handle **71** vertically at one position along time ruler **60**, the system automatically deforms line **70** along part of its length in accordance with a model in which the portion of line **70** to the left of the handle simulates an elastic band (to provide for smooth temporal variations in audio level assigned to consecutive segments of audio clip **67**). Thus, if the user drags a handle **71** downward from the position of pointer **68** in FIG. **9**, the system automatically moves downward a portion of line **70** on the left side of pointer **68**, as if that portion of line **70** were made of elastic material.

The user can create as many handles **71** as desired (three handles **71** are indicated in FIG. **9**). Each ascending portion of line **70** represents a fade in, and each descending portion of trace **70** represents a fade out. In response to entry of a preview command, the system retrieves from storage the audio data for audio clip **67** and plays the audio clip through a speaker (such as speaker **39** of FIG. **1**), with an instantaneous volume level determined by the position of line **70** relative to line **69** at the corresponding position along time ruler **60**.

Next, with reference to FIGS. **2**, **10**, and **11**, another way will be described in which a user can control superimposition of overlapping video (or still frame image) clips displayed in the construction window. To accomplish this, the system is programmed to superimpose a clip displayed in the "Superimpose" track (identified by label "Super" in FIGS. **2**, **10**, and **11**) with a clip displayed (along the same segment of time ruler **60**) in track A or B, or with overlapping clips displayed in tracks A and B. When the system generates and previews the corresponding program, the images corresponding to the clips in tracks A or B (or both A and B) shows through transparent parts of the superimposed image.

To superimpose a video or still image clip (referred to as an "overlay" clip) with a clip (referred to as a "main" clip) displayed in track A, B, or both A and B, the user drags the overlay clip to the "Super" track, and aligns it with the relevant portion of the clip in track A. For example, FIGS. **10** and **11** show a main clip **70** in track A and an overlay clip **72** in a "Super" track. The construction window of FIGS. **10** and **11** has been reconfigured to delete tracks B and FX. In a preferred embodiment, the user can reconfigure the construction window of FIGS. **10** and **11** to add tracks B and FX of the type shown in FIG. **2**.

The user specifies which portions of the overlay clip are transparent, in one of two ways. In one alternative, the user specifies a color (or range of colors) of the overlay clip as being transparent (so that the main clip will show through portions of the overlay clip having the specified color or range of colors). This can be done by displaying a transparency setting menu having icons which can be manipulated using mouse **38** to specify the "transparent" color or range. In another alternative, the overlay clip includes a special "alpha" channel, comprising bits which define a grey scale level for each pixel of each frame of the overlay clip. In this case, the system can be programmed to interpret the grey scale levels as degrees of transparency. An example of a clip having such an alpha channel is a clip consisting of 32-bit, still image frames generated by the "Photoshop" software commercially available from Adobe Systems Incorporated.

It is often desirable to generate an edited program having a title (or other overlay image) which fades in or out, or has a semi-translucent appearance, when superimposed on a main image. In a preferred embodiment, the system of the invention is programmed to enable the user to achieve such effects in a convenient, intuitive manner. This is accomplished by manipulating a level control icon, such as fade control bar **74** displayed in the "Super" track of FIGS. **10** and **11**. To adjust how "strongly" the overlay image will be displayed on the main image (or in other words, to adjust the weight with which overlay clip **72** is combined with main clip **73**), the user manipulates mouse **38** to position a pointer **76** on top (horizontal) line **74A** of fade control bar **74**. The user then "clicks" mouse **38** to display a handle **76A** (a black dot), and drags handle **76A** up or down within fade control bar **74** to deform a displayed "rubber-band" level control line **78**. The default position of rubber-band line **78** is a position coincident with top line **74A**, which represents maximum weight for overlay clip **72**. If the user drags handle **76A** downward (upward), the system assigns decreased (increased) weight to the corresponding segment of the overlay clip. With handle **76A** at the bottom of icon **74**, the corresponding overlay image will be completely non-visible when superimposed on the corresponding segment of main clip **73**.

Line **78** has "rubber-band" characteristics in the sense that, in response to a user's dragging of handle **76A** vertically at one position along time ruler **60**, the system automatically deforms line **78** along part of its length in accordance with a model in which the portion of line **78** to the left of the handle simulates an elastic band (to provide for smooth temporal variations in the weight assigned to consecutive segments of overlay clip **72**). Thus, if the user drags handle **76A** downward from the position shown in FIG. **11**, the system automatically moves downward a portion of line **78** on the left side of handle **76A**, as if that portion of line **78** were made of elastic material.

One advantage of the invention is that it provides a convenient and intuitive way for a user to select "In" and "Out" points for each clip to be included in an edited video program. In contrast, prior art editing systems accomplish this function in an inconvenient manner, in which the user views sequentially displayed video frames, freezes the sequential display at a desired frame, and marks the frame as an "In" or "Out" point of a video segment. This prior art technique makes it very difficult for a user to perform many routine editing operations, such as extending a video segment to match certain actions in another video segment (or to match a desired portion of an audio soundtrack).

The inventive system is preferably programmed to enable a user to select "In" and "Out" points for each clip of a video program, in the manner to be described next with reference to FIGS. **6** and **7**. To align a clip along time ruler **60**, the clip is displayed in a track of the construction window, and the left or right edge of the clip is dragged along the track to a desired "In" or "Out" point along the time ruler **60**. The precision with which each edge can be positioned along time ruler **60** depends on the currently selected time scale for the construction window (the time scale is selected, in the manner explained above, by manipulating icon **58**). To align clips with maximum precision, the time unit should be selected to have its lowest value (i.e., icon **58** configured in the "single frame" setting, in which each frame of the clip is displayed in the construction window).

As shown in FIG. **6**, left edge **81** of clip **80** is aligned at SMPTE time code 0:00:10:00 of time ruler **60**, indicating that clip **80** commences (i.e., has an "In" point) at the tenth second of the video program being edited. Right edge **82** of clip **80** is aligned at SMPTE time code 0:00:11:00 of time

ruler **60**, indicating that clip **80** ends at the eleventh second of the video program. To maintain this "In" point for clip **80**, and insert a longer-duration version of clip **80** into the video program, the user positions a "stretch" cursor **83** at right edge **82** of clip **80** (as shown in FIG. **6**). The user then "grabs" edge **82** by manipulating a control on mouse **38** and drags edge **82** toward the right along time ruler **60**. When edge reaches the desired new "Out" point along ruler **60** (i.e., the point marked with SMPTE time code 0:00:12:00, as shown in FIG. **7**), the user releases the control on mouse **38**. In response to this sequence of operations, the system edits the video program to include a longer version of clip **80** (having the original "In" point and the new "Out" point), and retrieves from memory additional frames of clip **80** for display in the construction window (i.e., the two new frames of clip **80** shown in FIG. **7**, but not FIG. **6**).

If the user enters the preview command with video clip **80** displayed as shown in FIG. **6**, the system would display (in a preview window) all frames of a one-second video segment. If the user enters the preview command with video clip **80** displayed as shown in FIG. **7**, the system would display (in the preview window) all frames of a two-second video segment (with the first one-second portion of such two-second segment identical to the segment referred to in the previous sentence).

Similarly, the user can set a new "In" point for a clip by dragging the clip's left edge, along a track in the construction window, with the stretch icon.

Also similarly, microprocessor **31** is preferably programmed to enable a user to split a clip into two new clips (which can be independently resized and repositioned along time ruler **60**) by positioning a "razor blade" cursor at desired position along the time ruler and manipulating a control on mouse **38** to define this desired position as the "Out" point of the first new clip and the "In" point of the second new clip.

Microprocessor **31** is preferably programmed to enable a user to select a special effect transition icon, in the following manner, for display in the above-described "FX" track of the construction window. In response to a command from the user, microprocessor **31** causes a special effects menu (such as the menu shown in FIG. **12**) to be displayed in a window on monitor **36**. The special effects menu includes a set of transition icons, each which can be selected by positioning a cursor thereon using a mouse.

Each transition icon represents a different special effect transition between two clips: an "A" clip (displayed in track A of the construction window, and associated with an "A" image); and a "B" clip (displayed in track B of the construction window, and associated with a "B" image). The following fourteen transition icons are shown in FIG. **12** (more than or less than fourteen transition icons can be displayed in alternative embodiments of the invention):

a "Band Slide" icon, representing a transition in which the B image appears to slide over the A image in horizontal or vertical bars;

a "Band Wipe" icon, representing a transition in which the B image is revealed under the A image by horizontal or vertical bars;

a "Barn Doors" icon, representing a transition in which the B image is revealed under the A image from the center outwards;

a "Checkerboard" icon, representing a transition in which boxes wipe in a spiral to reveal image B under image A;

a "Cross Dissolve" icon (also shown within icon **65** in FIG. **3**), representing a transition in which image A fades into image B;

a "Dither Dissolve" icon, representing a transition in which image A dissolves into image B;

a "Funnel" icon, representing a transition in which the A image is pulled through a funnel, revealing the B image;

an "Inset" icon, representing a transition in which a corner wipe reveals image B under image A;

an "Iris Round" icon, representing a transition in which a circular wipe opens to reveal image B under image A;

an "Iris Square" icon, representing a transition in which a rectangular wipe opens to reveal image B under image A;

a "Page Turn" icon, representing a transition in which image A curls to reveal image B underneath;

a "PICT Mask" icon, representing a transition in which a user selected 1-bit (black and white) image, of the conventional PICT type, and the system replaces the black in the PICT image with image A and the white in the PICT image with image B;

a "Push" icon, representing a transition in which image B appears to push aside image A; and

a "Radial Wipe" icon, representing a transition in which a line fixed at one corner of the screen sweeps across image A, revealing image B.

In response to a command from the user to activate the special effects menu, the system animates each of the transition icons displayed in the special effects menu. For example, FIG. **13** is a set of six transition icon displays, representing the animated display of the "Push" transition icon of FIG. **12**. The left-most image of FIG. **12** is displayed at a first time (t=0), followed (in sequence) by the second image from the left, the third image from the left, the fourth image from the left, the fifth image from the left, and the right-most image. After the right-most image is displayed, the animation process repeats, and all six images are sequentially displayed again.

The user includes a special effect transition in the program being edited by selecting a corresponding transition icon from the special effects menu, and dragging the selected transition icon to a desired position along the FX track of the construction window. As described above with reference to FIG. **3**, the user adjusts "in" and "out" points of the transition icon in the FX track so that the left edge ("in" point) and right edge ("out" point) of the transition icon are aligned with beginning and end points of a selected overlapping portion of a first clip in track A and a second clip in track B (as transition icon **65** is configured in FIG. **3**). In one class of embodiments, the user adjusts in and out points of each transition icon by positioning a cursor on the left or right edge of the icon, and dragging the edge relative to time ruler **60**, in the same manner (described above with reference to FIGS. **6** and **7**) that a user adjusts "in" and "out" points of a video or still image clip in track A or track B.

In a class of preferred embodiments, each transition icon includes a set of control icons for setting parameters of the corresponding special effect transition (such as the in point, out point, and duration of the transition). Icon **90** of FIG. **15** is an example of a transition icon having such control icons.

In FIG. **15**, the control icon displayed in area **80** determines the type of the special effect transition. The user determines the type of the transition by dragging a selected icon from a special effects menu (which can be the menu described above with reference to FIG. **12**) to area **80**.

The user can specify a direction of movement for certain transitions (i.e., the direction on the screen in which one clip "pushes" the other in a "push" transition) by "clicking" mouse **38** to select a desired one of edge control icons **82**.

If there are at least two available directions for a transition, the corresponding icon **90** displays one edge control icon **82** for each available position. For example, for a "push" transition (when a "push" icon of the type described with reference to FIG. **13** is displayed in area **80** of icon **90**), eight edge control icons **82** may be displayed (as shown in FIG. **15**), each representing a different direction in which one of clips A and B will push the other during the transition.

Icon **90** optionally includes track selection control icon **88**. The user can specify which clip (as between a clip in track A and an overlapping clip in track B) should play a given role in the transition, by clicking on icon **88** using mouse **38**.

Icon **90** optionally also includes forward/reverse control icon **84**, and anti-aliasing control icon **86**. The user can specify a forward or reverse direction for a transition by clicking on icon **84** using mouse **38**. The user can set an anti-aliasing level (i.e., to a "low," "high," or "off" level) by clicking on icon **86** using mouse **38**.

An additional feature of a preferred embodiment of the invention will be described next with reference to FIG. **14**. In this embodiment, microprocessor **31** is programmed to perform any of a variety of filtering operations on selected clips. The user specifies each filtering operation by displaying a corresponding filter control menu, and setting filtering parameters by manipulating filter control icons displayed on the filter control menu using mouse **38** and/or keyboard **40**.

An example of such a filter control menu (for a "mosaic" filter) is shown in FIG. **14**. A mosaic filter divides the frames of the displayed clip into a grid of squares, and makes each square the average color of all the pixels within the square. The user can set the effect of the mosaic filter to gradually increase or decrease as the clip plays by adjusting "Start" and "End" control icons displayed in the mosaic filter control menu. By lengthening "Start" control icon **92**, the user increases the level of the mosaic filter at the start of the clip (i.e., increases the size of the squares into which the first frame of the clip is divided). Similarly, by lengthening "End" control icon **94**, the user increases the level of the mosaic filter at the end of the clip (i.e., increases the size of the squares into which the last frame of the clip is divided). The system is preferably programmed to interpolate the level of the temporally varying mosaic filter for intermediate frames of the clip. When the user has defined a temporally varying mosaic filter for a clip, and then enters the preview command, the system filters the digital data corresponding to each frame of the clip (in accordance with the defined mosaic filter) before displaying the filtered clip in the preview window.

Another feature of a preferred embodiment of the invention will be described next. This feature contemplates that microprocessor **31** can be programmed to execute any of a variety of special effects on selected clips. One preferred way to accomplish this is to program microprocessor **31** with "plug-in" special effect software modules, each of which performs a phase of a more complicated special effect. This feature also contemplates that the frames of digital data to be processed include an alpha channel (i.e., an 8-bit, transparency-scale representation of an image for each frame) associated with color channels (which can comprise three 8-bit color values for each pixel of the frame).

When performing special effects processing (requiring combination of two clips), the programmed microprocessor **31** of the invention preferably fills the alpha channel of the first clip with a first value (i.e., a hexadecimal "00"), fills the alpha channel of the second clip with a second value (i.e., a hexadecimal "1F"), and then performs a first special effect process (i.e., in a plug-in special effect software module) in which selected pixels from the first clip are combined with selected pixels of the second clip to generate a processed clip. Then, when the processed clip undergoes subsequent processing, the alpha channel data of the processed clip is examined to identify one or more edges of the processed clip. For a given pixel, if the pixel's alpha channel matches the alpha channels in the four adjacent pixels (the pixels left, right, above, and below it) then there is no edge at that pixel. If, however, one or more of the alpha channels differs, an edge is determined to exist at that pixel (and any desired operation can be applied at that location, such as colored edge or a smoothing operation).

This technique allows the second plug-in special effect software module (or other software) to identify edges of the processed clip without the need to specifically generate an edge region data structure.

The choice of "00" and "1F" values allows for an additional speed increase in the comparison. Instead of comparison of the alpha channels as described above, the alpha channel values of the five neighboring pixels can be added together as byte sized values. A result of zero or a negative value indicates no edge, while any other value indicates an edge.

Next, the caching steps (briefly described above) implemented by a preferred embodiment of the invention will be more fully described. Data are written into the cache memory (i.e., random access memory **32** of FIG. **15**) in response to each user command for updating the construction window, or for previewing a program defined by a currently displayed construction window. An example of the latter type of command is a command to change the time scale of time ruler **60**.

In response to each such command, the system calculates the addresses of the data that must be retrieved to update the construction window (or a preview window for previewing the program). The system then searches the cache memory to retrieve therefrom any of the required data currently stored in the cache memory. After retrieving any of the required data currently stored in the cache memory, the system retrieves from its mass storage memory (memory **33** of FIG. **1**) the remaining required data. The system then caches all the retrieved data in the cache memory, processes the retrieved data in response to the command, and displays the processed data to update the screen display. As a result of the caching step, the system reduces the memory access time required for responding to future construction window updating or preview commands.

For example, in an embodiment of the invention, in response to a command for previewing a program, microprocessor **31** retrieves all frames of the program (from cache memory or mass storage memory), processes the retrieved frames in accordance with special effects transitions defined by transition icons currently positioned in the construction window (and in accordance with any user-specified filtering operation), and causes the processed frames to be sequentially displayed in a preview window on the screen of monitor **36**. The system also caches the retrieved frames (in their unprocessed form).

In preferred embodiment, microprocessor **31** "compresses" each frame to be cached in memory **32**, in the sense that it causes only some of the bits of the frame to be cached. For example, if a clip comprises 32-bit digital video data, including an eight-bit alpha channel for each frame of the clip, retrieved frames of the clip may be "compressed" by

eliminating the alpha channel before caching the remaining 24-bit digital video data defining each frame. Such an alpha channel can be employed by microprocessor **31** to perform certain special effects processing (described above), and accordingly, such special effects processing must be performed on the retrieved frames before the frames are "compressed."

Various other modifications and alterations in the structure and method of operation of this invention will be apparent to those skilled in the art without departing from the scope and spirit of this invention. Although the invention has been described in connection with specific preferred embodiments, it should be understood that the invention as claimed should not be unduly limited to such specific embodiments.

What is claimed is:

1. A method for editing a video program which includes a sequence of clips and transitions between the clips, the method comprising the steps of:

(a) displaying a time line;

(b) generating a display representing the video program clips by digitizing and thereafter simultaneously displaying a frame from each of a plurality of the digitized clips on tracks along and in a direct time relationship to the time line the width of the displayed frames representing the time duration of the clip represented by that frame, together with transition icons, the transition icons representing the transitions between the plurality of the digitized clips;

(c) editing the video program by modifying the display.

2. The method of claim **1**, wherein step (c) includes the step of:

dynamically generating and displaying the frames in digital form in response to commands for modifying the display.

3. The method of claim **2**, wherein each of the frames has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein step (c) includes the step of:

changing an in point of a first one of the clips by changing a position along the time line of the left edge of the frame representing the first one of the clips.

4. The method of claim **2**, wherein each of the has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein step (c) includes the step of:

changing an out point of one of the clips by changing a position along the time line of the right edge of the frame representing one of the clips.

5. The method of claim **1**, wherein a first one of the transition icons is displayed in a special effects track, and has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein step (c) includes the step of:

changing the in point of the first one of the transition icons by changing a position along the time line of the left edge of said first one of the transition icons.

6. The method of claim **1**, wherein a first one of the transition icons is displayed in a special effects track, and has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein step (c) includes the step of:

changing the out point of the first one of the transition icons by changing a position along the time line of the right edge of said first one of the transition icons.

7. The method of claim **1**, wherein step (c) is performed in response to a command for modifying the display, and wherein step (c) includes operations of:

(d) identifying data needed to modify the display in response to the command;

(e) retrieving from a cache memory at least a first portion of the data currently stored in the cache memory;

(f) after step (e), retrieving any remaining portion of the data from a mass storage memory;

(g) processing the data retrieved during steps (e) and (f) to generate processed data for modifying the display in response to the command; and

(h) caching the data retrieved during steps (e) and (f) in the cache memory by compressing the data and caching the resulting compressed data in the cache memory.

8. The method of claim **1**, wherein the display includes a first frame in digital form displayed on a first track, a second frame in digital form displayed on a second track with an overlap portion of the second frame aligned along the time line with an overlap portion of the first frame, and a transition icon displayed in a third track in alignment with the overlap portion of the second frame.

9. The method of claim **8**, wherein the transition icon has an associated control icon, and wherein step (c) includes a step of:

modifying a parameter of a transition in the video program between the first clip and the second clip in response to manipulation of the control icon using an input device.

10. The method of claim **8**, wherein the transition icon is an animated icon which simulates a transistion in the video program between the first clip and the second clip.

11. The apparatus of claim **8**, wherein the transition icon is an animated icon which simulates a transition in the video program between the first clip and the second clip.

12. The method of claim **1**, wherein step (c) adds to the video program a special effect transition between a first clip and a second clip, and wherein step (c) includes the steps of:

(d) displaying a first frame on a first track, and a second frame on a second track with an overlap portion of the second frame aligned along the time line with an overlap portion of the first frame;

(e) selecting a first transition icon; and

(f) positioning the first transition icon in a third track in alignment with the overlap portion of the second frame.

13. The method of claim **12**, wherein step (e) includes the steps of displaying a menu of transition icons including the first transition icon and selecting the first transition icon in response to manipulation of an input device, and wherein step (f) includes a step of dragging the first transition icon from the menu to the third track in response to manipulation of the input device.

14. The method of claim **13**, wherein each of the transition icons of the menu is an animated icon which simulates an available transition in the video program.

15. The method of claim **1**, wherein step (c) includes the steps of:

displaying a filter menu having control icons representing parameters of a temporally varying filter for filtering a selected one of the clips; and

determining the parameters of the temporally varying filter in response to manipulation of the control icons using an input device.

16. The method of claim **15**, wherein step (c) includes the step of:

filtering the selected clip with the temporally varying filter.

17. The method of claim **15**, wherein the temporally varying filter is a mosaic filter, and the parameters include a start level and a final level for the mosaic filter.

18. The method of claim 1, wherein step (c) includes a step of performing special effects processing on a first clip and a second clip, said step of performing special effects processing including the steps of:

filling an alpha channel of the first clip with a first value;

filling an alpha channel of the second clip with a second value;

performing a first special effect process in which selected data of the first clip are combined with selected data of the second clip to generate a processed clip having an alpha channel; and

processing data in the alpha channel of the processed clip to identify one or more edges of said processed clip.

19. The method of claim 18, wherein the first value is a hexadecimal 00 value, and wherein the second value is a hexadecimal 1F value.

20. An apparatus for editing a video program which includes a sequence of clips which have been digitized, each represented by a frame, the width of each frame representing a time duration of the digitized clip represented by that frame, and transitions between the clips, the apparatus comprising:

a monitor;

a mass storage memory for storing the digitized clips;

a processing means coupled to the monitor and the mass storage memory, wherein the processing means is programmed with:

software for displaying a time line,

software for generating a display representing the video program on the monitor, the display including a plurality of the frames displayed simultaneously on tracks in a direct time relationship to the time line, together with transition icons representing the transitions between the clips, and

software for editing the video program by modifying the display.

21. The apparatus of claim 20, also including:

an input device coupled to the processing means,

and wherein the processing means is programmed with software for dynamically generating and displaying new frames in response to commands from the input device for modifying the display.

22. The apparatus of claim 21, wherein each of the frames representing the clips has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein the processing means is programmed to change an in point of a first of the clips by changing a position along the time line of the left edge of a first frame.

23. The apparatus of claim 21, wherein each of the frames has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein the processing means is programmed to change an out point of a first of the clips by changing a position along the time line of the right edge of a first frame.

24. The apparatus of claim 21, wherein the display includes a first frame displayed in a first track, a second frame displayed in a second track with an overlap portion of the second frame aligned along the time line with an overlap portion of the first frame and a transition icon displayed in a third track in alignment with the overlap portion of the second frame.

25. The apparatus of claim 24, wherein the transition icon has an associated control icon, and wherein the processing means is programmed with software for modifying a parameter of a transition in the video program between the first clip and the second clip in response to manipulation of the control icon using the input device.

26. The apparatus of claim 21, wherein the processing means is programmed with software for adding to the video program a special effect transition between a first digitized clip and a second digitized clip, in response to steps of:

manipulating the input device to display a frame representing the first digitized clip in a first track, and a second frame representing the second digitized clip in a second track with an overlap portion of the second frame aligned along the time line with an overlap portion of the first frame;

manipulating the input device to select a first transition icon; and

manipulating the input device to position the first transition icon in a third track in alignment with the overlap p portion of the second frame.

27. The apparatus of claim 26, wherein the processing means is programmed with:

software for displaying a menu of transition icons including the first transition icon; and

software for selecting the first transition icon in response to manipulation of the input device.

28. The apparatus of claim 27, wherein each of the transition icons of the menu is an animated icon which simulates an available transition in the video program.

29. The apparatus of claim 21, wherein the processing means is programmed with:

software for displaying a filter menu having control icons representing parameters of a temporally varying filter for filtering a selected digitized clip; and

software for determining the parameters of the temporally varying filter in response to manipulation of the control icons using the input device.

30. The apparatus of claim 29, wherein the processing means is programmed with software for filtering the selected digitized clip with the temporally varying filter.

31. The apparatus of claim 29, wherein the temporally varying filter is a mosaic filter, and the parameters include a start level and a final level for the mosaic filter.

32. The apparatus of claim 20, wherein a first one of the transition icons is displayed in a special effects track, and has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein the processing means is programmed to change the in point of the first one of the transition icons by changing a position along the time line of the left edge of said first one of the transition icons.

33. The apparatus of claim 20, wherein a first one of the transition icons is displayed in a special effects track, and has a left edge aligned at an in point along the time line and a right edge aligned at an out point along the time line, and wherein the processing means is programmed to change the out point of the first one of the transition icons by changing a position along the time line of the right edge of said first one of the transition icons.

34. The apparatus of claim 20, also including:

a cache memory coupled to the processing means, and wherein the processing means is programmed with software for identifying data needed to modify the display in response to each of the commands for modifying the display, then retrieving from the cache memory at least a first portion of the data currently stored in the cache memory, then retrieving any remaining portion of the data from the mass storage memory,

then processing the retrieved data to generate processed data for modifying the display, and caching the retrieved data in the cache memory.

35. The apparatus of claim **20**, wherein the processing means is programmed with software for performing special effects processing operation on a first digitized clip and a second digitized clip the operation including the steps of:

filling an alpha-channel of the first digitized clip with a first value;

filling an alpha channel of a second digitized clip with a second value;

performing a first special effect process in which selected data of the first digitized clip are combined with selected data of the second digitized clip to generate a processed clip having an alpha channel; and

processing data in the alpha channel of the processed clip to identify one or more edges of the processed clip.

36. The apparatus of claim **35**, wherein the first value is a hexadecimal 00 value, and wherein the second value is a hexadecimal 1F value.

* * * * *

US006686918B1

(12) **United States Patent**
Cajolet et al.

(10) **Patent No.: US 6,686,918 B1**
(45) **Date of Patent: Feb. 3, 2004**

(54) **METHOD AND SYSTEM FOR EDITING OR MODIFYING 3D ANIMATIONS IN A NON-LINEAR EDITING ENVIRONMENT**

(75) Inventors: **Claude Cajolet**, Montreal (CA); **Réjean Gagné** , Montreal (CA)

(73) Assignee: **Avid Technology, Inc.**, Tewksbury, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/049,066**

(22) Filed: **Mar. 27, 1998**

(51) **Int. Cl.**[7] **G06T 13/00**

(52) **U.S. Cl.** **345/473**; 345/419

(58) **Field of Search** 345/473–475, 345/302, 419

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,261,041	A *	11/1993	Susman	345/473
5,781,188	A	7/1998	Amiot et al.	
5,826,102	A *	10/1998	Escobar et al.	345/302
5,838,938	A *	11/1998	Morgan	345/302
6,011,562	A	1/2000	Gagne et al.	
6,023,703	A *	2/2000	Hill	345/302
6,058,397	A *	5/2000	Barrus et al.	345/302
6,064,393	A *	5/2000	Lengyel et al.	345/419 X
6,243,706	B1	6/2001	Moreau et al.	

FOREIGN PATENT DOCUMENTS

GB	2 317 292 A	3/1998
WO	WO 97/46975	11/1997

OTHER PUBLICATIONS

Adobe Systems, Inc., "Feature Highlights", Adobe After Effects, Mar. 1997.*

Corel Corporation , "Corel Draw 4", pp. 418, 454–456, 1993.*

U.S. patent application Ser. No. 09/063,289, Gagne, filed Apr. 21, 1998.

Adobe Systems Inc., "Feature Highlights", Adobe After Effects, Mar. 1997, pp 1–10.

Adobe Systems Inc., "The Most Powerful Tool For Digital Compositing, 2D Animation, and Special Effects", Adobe After Effects—Brochure, Mar. 1997.

Marc A. Najork and Marc H. Brown, "Obliq–3D: A High–Level, Fast Turnaround 3D Animation System" IEEE Transactions On Visualization and Computer Graphics, Vol 1, No. 2., Jun. 1995, pp 175–193.

Interactive Multimedia, vol. 2, Issue 3, The Integrator: A Prototype For Flexible Development Of Interactive.

Digital Multimedia Applications, Antonio Siocchi, et al, pp. 5–26, Jul.–Sep. 1991.

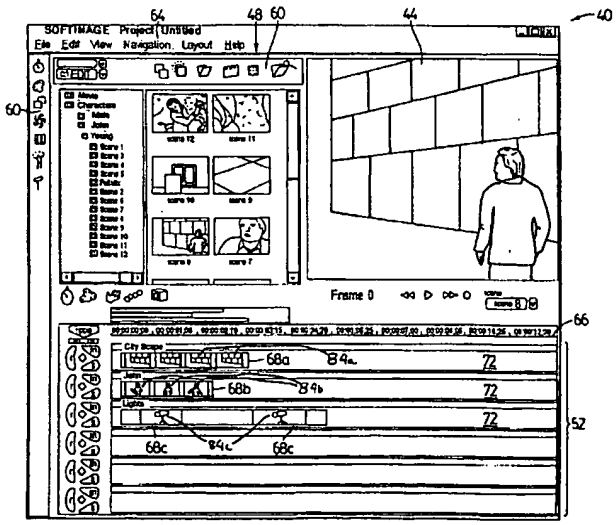
* cited by examiner

Primary Examiner—Matthew Luu
(74) *Attorney, Agent, or Firm*—Peter J. Gordon

(57) **ABSTRACT**

A non-linear editing (NLE) system for editing and/or modifying 3D animation information comprises elements represented by clip objects which can be positioned and/or manipulated relative to a time line. The elements can comprise conventional 1D (audio) or 2D (video) information or can comprise 3D animation information which can include animation objects and animation parameters associated with them, as well as 2D renderings of those objects. Positioning and/or manipulation of clip objects representing 3D animation elements can result in alteration of the 2D rendering and or re-rendering of the 3D information, as appropriate.

22 Claims, 5 Drawing Sheets



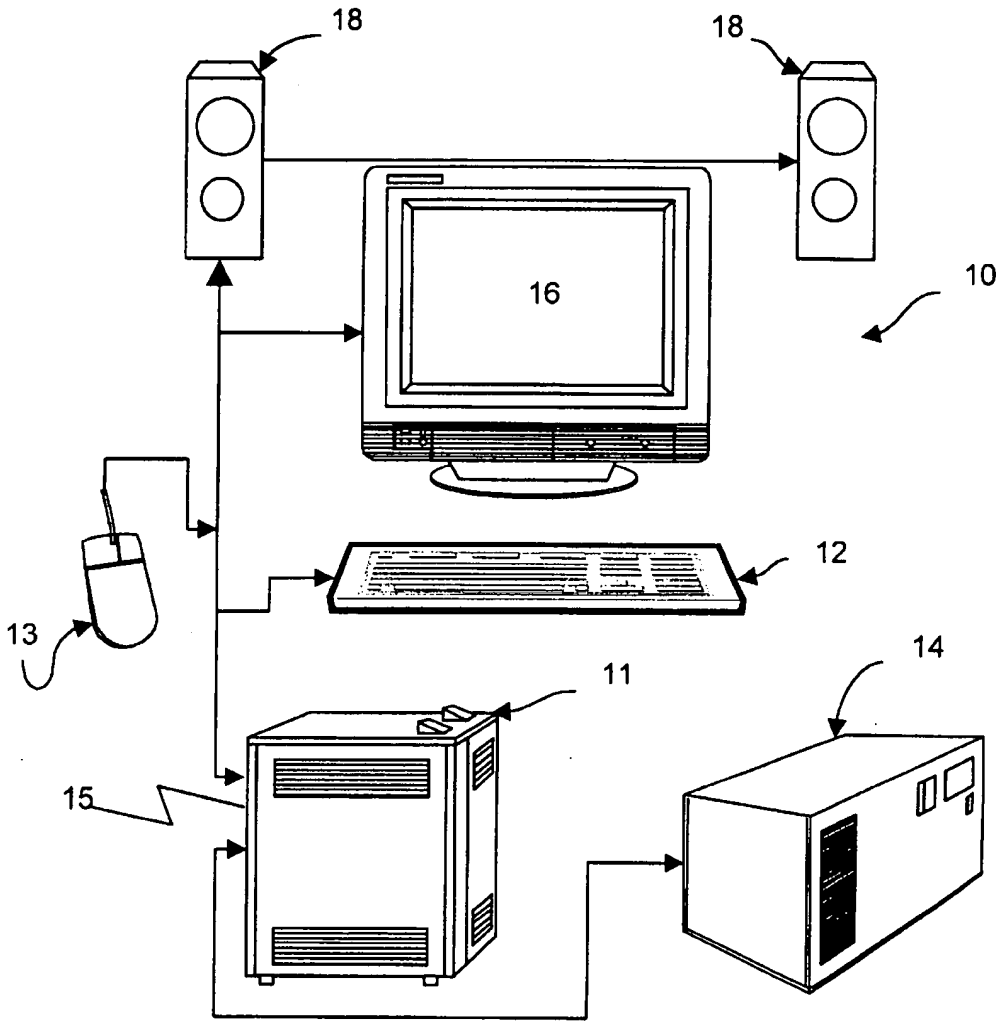


Fig.1

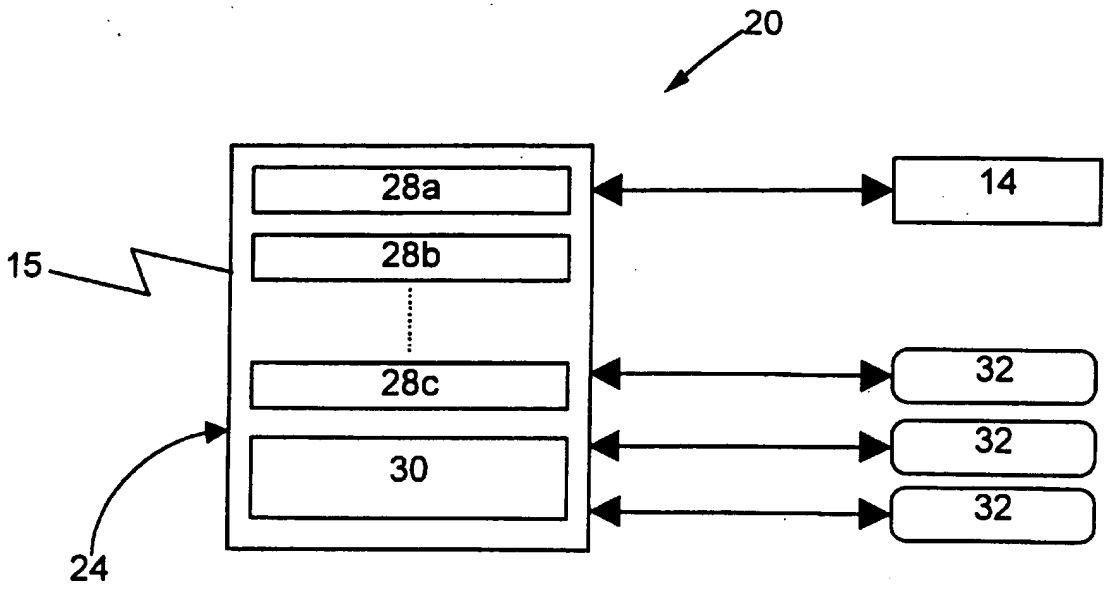

20
28a
28b
28c
30
14
32
32
32
15
24

Fig.2

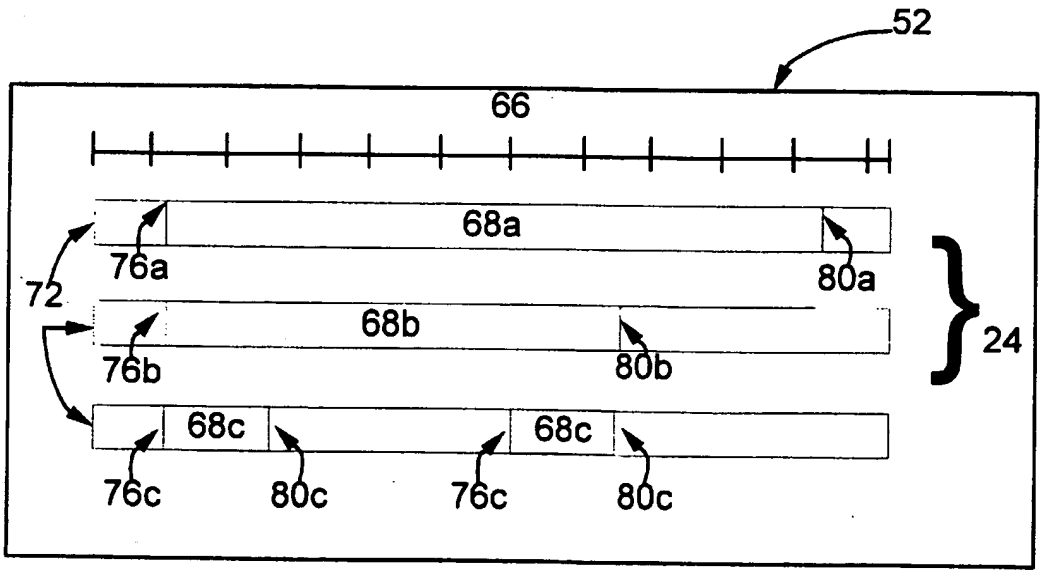
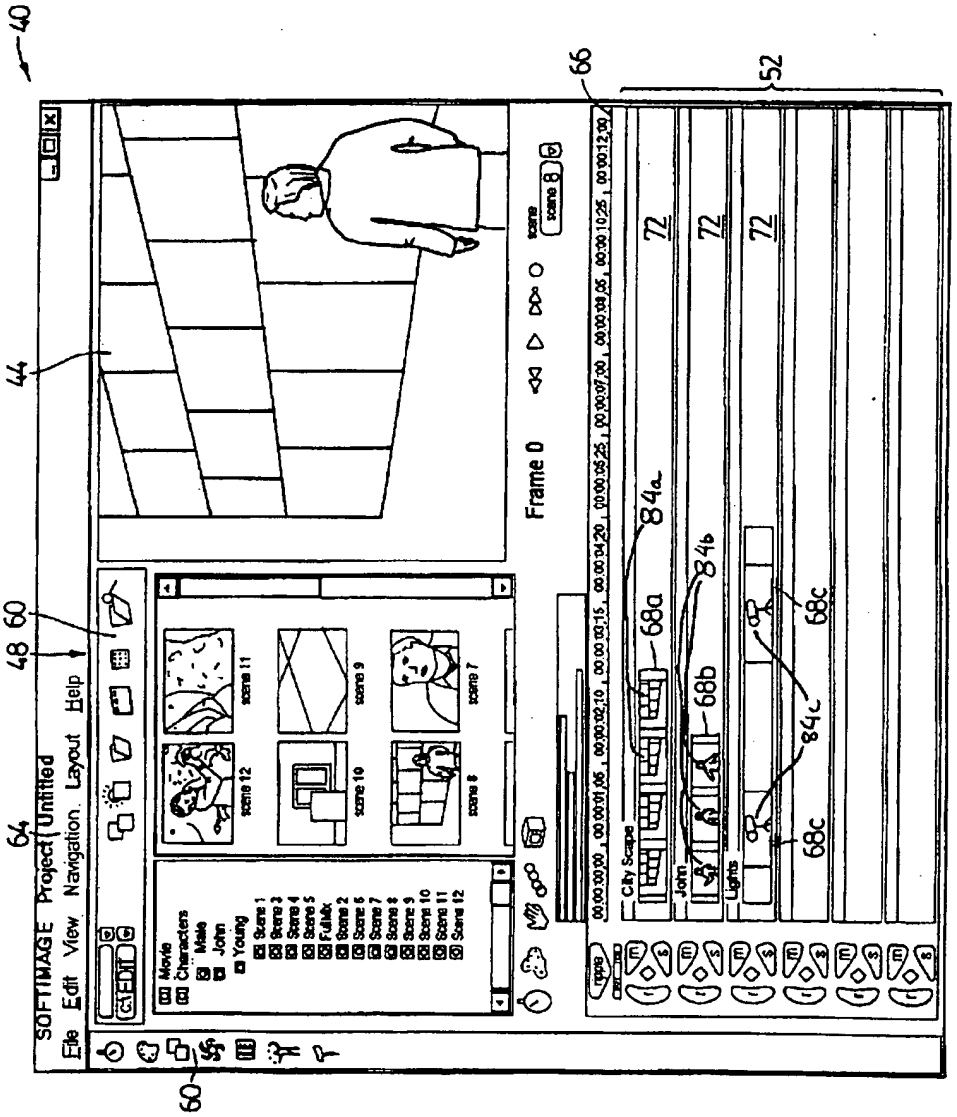

52
66
68a
76a
80a
72
68b
76b
80b
24
68c
68c
76c
80c
76c
80c

Fig. 4

FIG. 3


40
44
48
60
64
66
52
72
72
72
84a
84b
84c
68a
68b
68c
68c
60
SOFTIMAGE Project Untitled
File Edit View Navigation Layout Help
Frame 0
City Scape
John
Lights

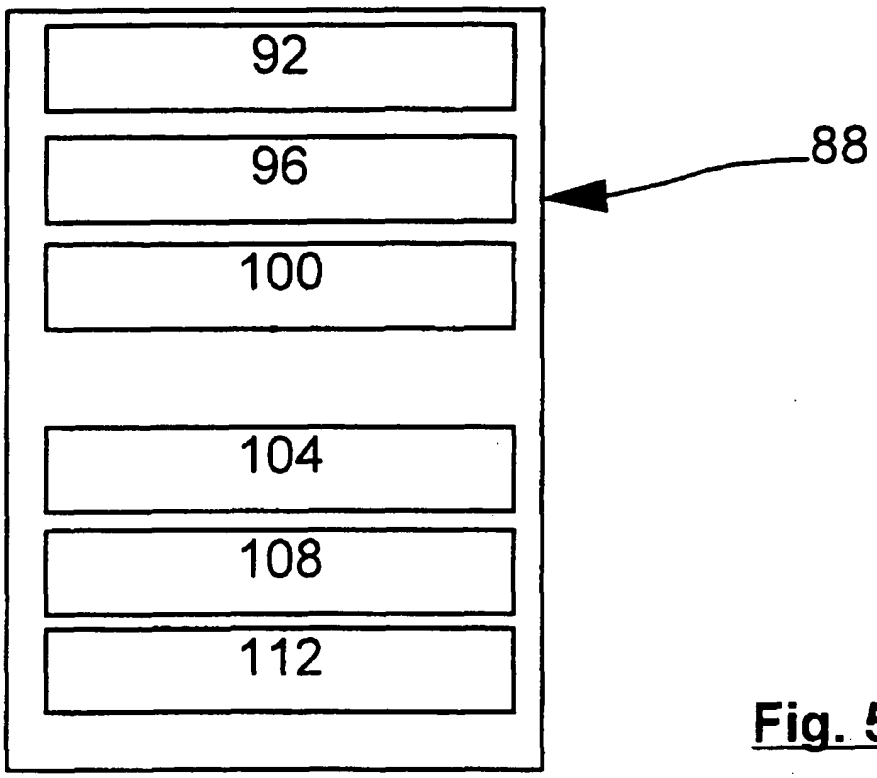


Fig. 5

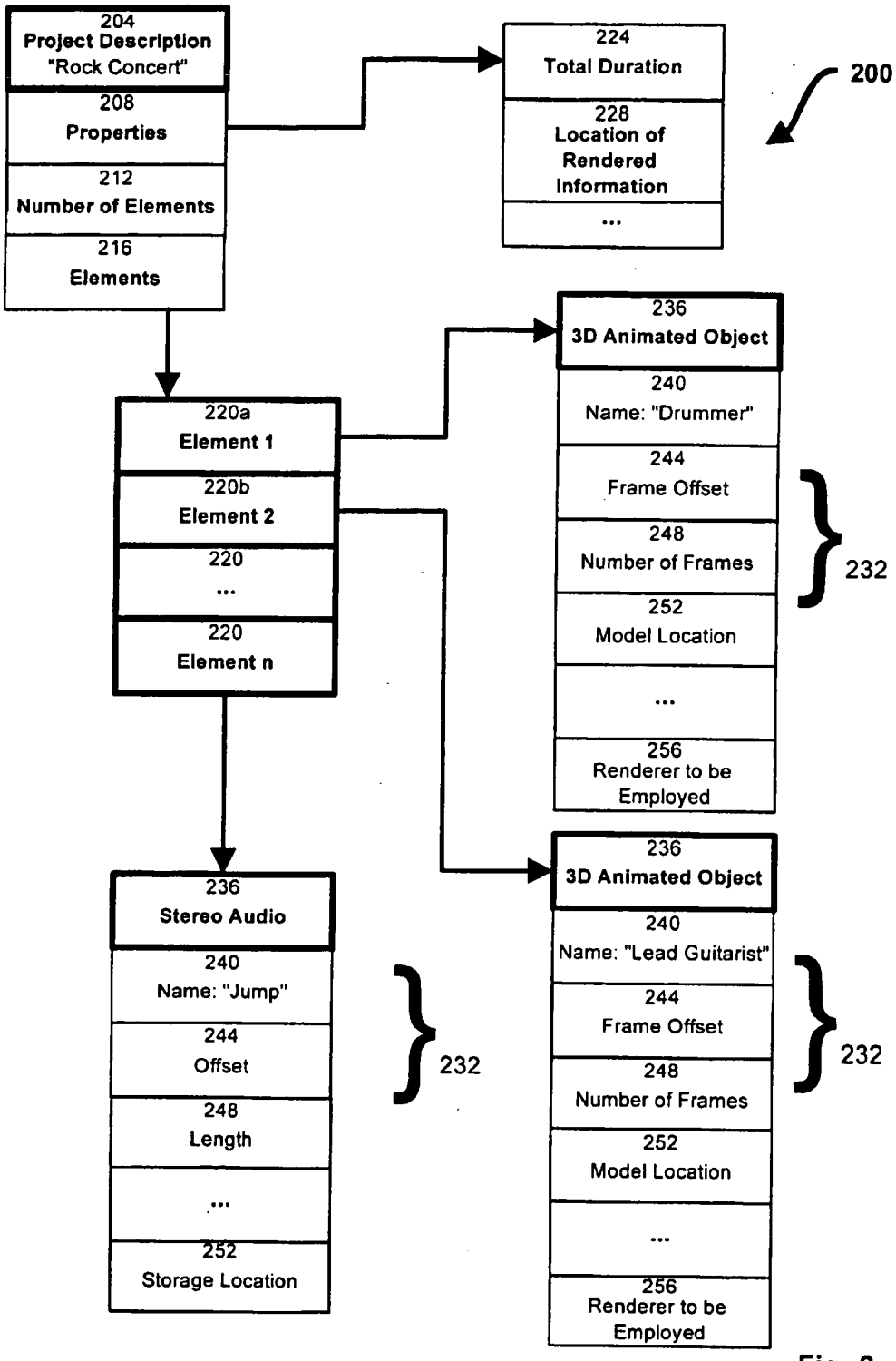

204
Project Description
"Rock Concert"
208
Properties
212
Number of Elements
216
Elements
224
Total Duration
228
Location of Rendered Information
...
200
220a
Element 1
220b
Element 2
220
...
220
Element n
236
3D Animated Object
240
Name: "Drummer"
244
Frame Offset
248
Number of Frames
252
Model Location
...
256
Renderer to be Employed
232
236
Stereo Audio
240
Name: "Jump"
244
Offset
248
Length
...
252
Storage Location
232
236
3D Animated Object
240
Name: "Lead Guitarist"
244
Frame Offset
248
Number of Frames
252
Model Location
...
256
Renderer to be Employed
232

Fig. 6

METHOD AND SYSTEM FOR EDITING OR MODIFYING 3D ANIMATIONS IN A NON-LINEAR EDITING ENVIRONMENT

FIELD OF THE INVENTION

The present invention relates to a system and method for editing or modifying 3D animations. In particular, the present invention relates to a non-linear editing system and method for editing and/or modifying 3D animations.

BACKGROUND OF THE INVENTION

Non-linear editing (NLE) systems are known for video (2D) and audio (sometimes referred to as "1D") systems. NLE systems for video and/or audio include systems manufactured by AVID, Quantel and others. Non-linear video editing systems allow a user to join, arrange and/or modify digital or digitized source information to create a final cut, or edit, for rendering to appropriate storage media or output. The systems are non-linear in that the sources are not generally limited to being accessed in a linear manner, but instead permit random access. For example, the last three seconds of a ten second digital video source can be accessed without accessing the first seven seconds, or individual frames in the digital video can be accessed and/or combined as desired. Non-linear audio editing systems allow a user to easily and efficiently manipulate digital or digitized audio sources to produce a desired final output, often referred to as a final edit or "cut".

In practice, an NLE system for video can, for example, be used to construct the video portion of a television advertisement for a new model of car, by connecting digitized source video of the new car driving on a country road, digital source video of a driver in the car and digitized source video of the car in a wind test tunnel with engineers standing about it. An NLE system for audio can be used to create an audio soundtrack for this advertisement, comprising a selection of classical music, in digital form, mixed with a narrator's voice-over, also in digital form, discussing the features of the car and various sound effects sources which are also mixed, at appropriate time points, with the music and narration sources. The video and audio final edits are then rendered to an appropriate sink, such as a VTR, RAID array, or a monitor, to obtain the finished commercial.

NLE editors represent the available sources (video or audio) as objects in a graphical user interface (GUI), the objects commonly being referred to as clips, which are positioned and/or manipulated with respect to a time line for the edit. Various effects, such as fades, dissolves, wipes, blends, etc. can be applied to the clips to obtain the final edit.

Known techniques for editing 3D animations have comprised the rendering of the 3D animation to 2D media (video) which is then edited in a conventional manner, such as with the above-described NLE. The 2D media rendered from the 3D animation is treated as any other 2D source material, such as digital video, and can be combined with such other source material, as desired by the director.

If the director requires a change to the 2D rendering of the 3D animation, for example to have a character walk past a fixed background at a faster rate, or to have the contents of the 3D animation composited together in a different order, the director must instruct an animation artist as to what is desired and the animation artist will then invoke the required animation tools to modify the animation and to re-render it to 2D media. This new 2D material would then be provided to the director to replace the previous material.

While these techniques have provided reasonable results in the past, it is desired to have a method and system of editing and/or modifying 3D animations which is easier to employ, more efficient than prior art editing techniques and which provides a flexible creative environment for directors, editors and artists.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a novel system and method of editing and modifying 3D animations.

The present invention provides a method and system for editing and modifying animation information which reduces or removes the barriers between a director and the source material he is working with. Conventionally, with 2D video, which can include 2D renderings of 3D animations, NLE systems are used to edit existing source information. Essentially, the director creating the final edit is only able to select, arrange and apply effects to source information which has already been created. If additional source information is required, the director must undertake to have such additional information created, by filming new scenes, etc. and there is thus a barrier between the director and his work. The present inventors have determined that an NLE which is used to edit source material from 3D animations need not be so limited and that the barrier experienced with such conventional systems can be mitigated. Specifically, the present inventors have created a system and method for the editing and modification of 3D animations which allows an NLE system to integrate the 2D representations which the director is dealing with to the 3D world of the animations. As described below, with the present invention the director or other user of the NLE system can edit and manipulate both 2D renderings of 3D animations in a conventional manner and can also edit, manipulate and re-render those 3D animations as necessary.

According to a first aspect of the present invention, there is provided a computer implemented method for creating and modifying an edit comprising at least 3D animation information, comprising the steps of the user selecting an element from a set of available elements, the element representing 3D animation information which includes at least one parameter, such as a start position, end position, animation speed, etc. The selected element is represented to the user as a clip object which the user can position in a graphical display of the computer relative to a time line. The size of the clip object in the graphical display, relative to the time line corresponds to a duration which is applied to appropriate ones of the parameters in the element and the edges of the clip object relative to the time line represent a start time and an end time each of which are also applied to appropriate ones of the parameters of the element. The computer accesses the animation information, in accordance with the information applied to the parameters, to create the edit by compositing together each element. Modifications to the edit can be achieved by the user repositioning and/or resizing the clip objects, the computer applying the corresponding changes to the appropriate parameters and re-compositing the edit, and by removing, adding or substituting elements which will be composited into the edit.

According to another aspect of the present invention, there is provided a computer implemented non-linear editing system for creating and modifying edits comprising at least 3D animations, comprising: a storage device to store elements including animation information and parameters which relate to the animation information; a computer operatively connected to the storage device to access the

elements; one or more output devices to display to a user a graphical user interface of an NLE, including a time line, and to display the result of an edit produced with one or more of the stored elements which are composited together to form the edit. The system also includes at least one user input device with which the user can: select at least one element from a list of said elements stored on said storage device; define the positioning, relative to the time line, of a clip object representing the selected element, the start time, end time and duration of the animation produced from the element being determined by the positioning and sizing of the clip object relative to the time line. The computer accesses each element in correspondence with the positioning and sizing of its respective clip object, and composites the elements together to produce the edit. When a user repositions and/or resizes one or more clip objects, the changes to the start time, end time and duration of the elements corresponding to the changes made to the clips are applied the appropriate parameters of the elements and the computer re-accesses and composites the elements together according to the new values of the parameters.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example only, with reference to the attached Figures, in which:

FIG. **1** shows a representation of an NLE system in accordance with the present invention;

FIG. **2** shows a block diagram of a data structure of the NLE system of FIG. **1**;

FIG. **3** shows a graphical user interface of the system of FIG. **1**;

FIG. **4** shows an enlarged schematic view of the time line area of FIG. **3**;

FIG. **5** is a block diagram of an animation element in accordance with the present invention; and

FIG. **6** is a block diagram of a project data structure.

DETAILED DESCRIPTION OF THE INVENTION

FIG. **1** shows an example of an NLE system **10** in accordance with an embodiment of the present invention. System **10** includes a computer which, for example, can be a general purpose computer system **11** such as a PC with an Intel Pentium processor and executing Microsoft Windows NT operating system, or a special purpose computer system, and which execute a series of program steps to provide an NLE-based system for editing and modifying 3D animations. Computer system **11** is connected to one or more user input devices, such as keyboard **12**, mouse **13**, or any other suitable user input device such as a graphics tablet (not shown), etc. While computer system **11** will generally include a non-volatile storage device, additional data storage for storing source information, edits in progress and final edits can be provided by a storage device such as RAID array **14**, optical disc drives (not shown), digital or analog video or audio tape recorders, etc.

As will be apparent to those of skill in the art, computer system **11** can be directly connected to storage devices, such as RAID array **14**, and/or be connected to storage devices via a suitable communications link, such as LAN network connection **15**, via the internet, etc. System **10** also includes one or more output devices, such as high resolution monitor **16**, sound system **18**, etc. and their related I/O cards (not shown). In the embodiment of FIG. **1**, the graphical user interface (GUI), described further below, of the NLE system and the outputs of the edit being produced are each displayed, either simultaneously or alternately, on monitor **16** with any associated sound track being output by sound system **18**. It is also contemplated however, that in some circumstances more than one monitor **16** may be provided to allow output to be displayed on one or more monitors while the GUI of the NLE system is displayed on another.

FIG. **2** is a block diagram of the data relationships **20** of a non-linear editing (NLE) system, in accordance with the present invention. In the following discussion, an edit created by NLE system **10** is referred to as a project **24**. A project **24** comprises a series of elements **28**, such as video, audio, still image, static 3D information and/or 3D animations which can be combined within project **24** and then rendered, mixed and/or composited to produce the final edit. In the Figure, three elements **28** are shown: first element **28***a* can be audio information, second element **28***b* can be information for a still image and third element **28***c* can be information for a three-dimensional animated character. As is indicated in the Figure, project **24** can include more than three elements **28** and can include, for example, multiple elements of audio, video and/or 3D animation, still images and static 3D objects. Further, as will be described below, elements **28** of project **24** can also themselves be projects. Further, each element **28** can also have one or more defined associations with appropriate tools **32**, as discussed in more detail below. In addition to elements **28**, project **24** includes an edit data structure **30** which maintains the interrelationships, defined by the user with system **10**, between elements **28** to produce the final edit and which maintains the associations between elements **28** and the appropriate tools **32**.

Project **24** can include elements **28** which utilise source information from storage devices, such as RAID Array **14**, digital video or audio tapes, etc., and project **24** can itself be stored on storage devices such as RAID Array **14**, storage devices connected to computer system **11** by network connection **15**, etc.

Tools **32** can include an animation program, such as the SoftImage|3D product sold by the assignee of the present invention, or components thereof, compositing systems, one or more rendering engines, such as the "mental ray" rendering engine included with the above-mentioned SoftImage|3D product, etc. As is known by those of skill in the art, the choice of rendering engine depends on the desired quality, speed, resolution and other related factors and those of skill in the art will understand the similar considerations to be made in selecting other tools **32**. As mentioned above, each element **28** can have an association with one or more appropriate tools **32**. For example, an element **28** which includes animation information for a 3D animated character can have a defined association to the animation component of the above-mentioned SoftImage|3D product which allows system **10** to invoke tool **32**, as necessary, to modify the animation information. These Associations can be defined in a variety of manners as will occur to those of skill in the art and can, for example, be defined according to extensions to filenames storing the information, defined according to storage locations, i.e.—a subdirectory can be provided for animations, a different subdirectory can be provided for still images, defined explicitly for the information by the user, etc.

A user interacts with NLE system **10** through an NLE graphical user interface (GUI) **40**, an embodiment of which is shown in FIG. **3**. As shown, GUI **40** consists of an imaging area **44**, function area **48** and an NLE time line area **52**.

Imaging area **44** can be a portion of a monitor screen or it can be a separate monitor, such as a high resolution monitor and imaging area **44** can display the contents of visual (video, still image, animations, etc.) clips, or portions of those clips, the visual output of the complete current edit, etc. Similarly, sound system **18** can output the contents of audio (music, narration, sound effects, etc.) clips, or portions of those clips, the audio output of the complete current edit, etc. In this manner, a director can review the contents of clips, crop or otherwise edit them as appropriate (described below) and view the results.

Function area **48** consists of, for example, function buttons **60** and function menus **64** associated with a particular tool **32**, if such a tool is active, or with system **10**. It is contemplated that, in many circumstances wherein a tool **32** is accessed to modify a clip, via the association defined between the clip and the tool **32**, function area **48** will be updated to display at least a subset of the commonly used functions and operators of that tool **32**. For example, in an animation application, the functions can include shading, scaling, predefined movements, and other effects, as well as general file retrieval and the like.

If no specific tool **32** has been invoked, function area **48** displays functions and operators relating to system **10** and this can include an Explorer-style browser, as shown in the Figure, which allows a user to select projects of interest, such as the illustrated Scenes **7** through **12**, or available clips to be operated on with system **10**.

NLE time line area **52** is shown schematically in FIG. **4** and thumbnails **84** have been omitted for clarity. NLE time line area **52** displays a project **24** for "Scene **8**" of FIG. **3** in graphical form in relation to a time line **66** where time advances from left to right. Elements **28** are displayed as separate rectangular clips **68**, each in a track **72**. As described above for elements **28**, clips **68***a*, **68***b* and **68***c* can comprise audio information, video information, still image information (such as a background scene) or static 3D information (such as a 3D set through which an animated character walks) and animated 3D object information, such as an animated character or other model, respectively. In the specific time line for the project "Scene **8**" shown in FIGS. **3** and **4**, clip **68***a* represents an element **28***a* of a still image background, clip **68***b* represents an element **28***b* of an animated 3D character, in this case a man, and clip **68***c* represents an element **28***c* of a light source.

In this specific example, element **28***a* can have an association defined with a paint tool **32**, to allow an editor to modify the still image, and to a rendering engine tool **32** so that element **28***a* can be included by the rendering engine tool, should it become necessary to re-render the project as a result of changes to elements **28** made within system **10**.

Each clip **68**, and the underlying element **28** that it is associated with, have certain features in common. Each clip **68** is represented by a two-dimensional rectangular box, and each has, in relation to time line **66**, a start point **76** and an end point **80** defining a duration therebetween. The start point **76**, end point **80** and duration of each clip **68** is mapped to appropriate points in the corresponding element **28** which can be larger in total duration than the duration specified by its corresponding clip **68**. Each track **72** can have more than one clip **68**, or more than one instance of the same clip, as shown in the Figures wherein two clips **68***c* are shown in one track. When two instances of a clip are included, they can be either repeated, overlapping or different portions of the total duration of the element **28** associated with the clip **68**.

While the concepts of NLE-type editors are known for video and audio, the present invention employs the NLE metaphor to edit and modify 3D animation information, effectively providing a convenient and efficient 2D link to 3D information.

As shown in FIG. **3**, in a presently preferred aspect of the invention each clip **68** can also have associated with it thumbnails **84** which are preferably graphic representations, or icons, representing the content of the clip, although they can also be a textual description or both. When graphical representations or icons are employed, these thumbnails **84***a*, **84***b*, **84***c* can indicate the change or evolution in the underlying information over time, if any. For example, the thumbnail **84***a*, which shows an icon of a background, represents the information in element **28***a*. The information in clip **28***a* does not change over time, therefore, a single repeating thumbnail is sufficient to represent the information. By contrast, thumbnails **84***b* on clip **68***b*, are changing icons representing the state of the information in element **28***b*, which is associated with a 3D animation of the character "John" who walks as the animation proceeds. Clip **68***c* represents an element **28***c* which, in this example, comprises information relating to a light source to be included in the rendering of the project and, in this specific example, thumbnail **84***c* occurs twice along time line **66** so that the light source has two separate durations. This has the effect that the light is "on", i.e.—included in any rendering of the animation, for a first duration within the edit and "off", i.e.—not included in the rendering of the animation, for a subsequent duration within the edit and then "on" again for another duration of the edit.

To produce an edit and/or to modify the information in an element **28**, a user modifies or operates upon the corresponding clips **68** in the NLE time line area **52**. A user can select elements to be included in an edit from a list of available elements presented in the browser of function area **48**. The user can drag a desired element from this list and drop it onto a track **72** where it will be represented by a clip **68**. A user can change the duration of a clip **68** by scaling clip **68** to the desired length. This can be accomplished in a wide variety of manners, as will be understood by those of skill in the art, including well known "dragging" operations wherein a user clicks on either the start or ending edge of clip **68** with an input device such as mouse **13** and drags the edge to the desired new position. Similarly, a clip **68** can be moved along its associated track **72** to change its start point **76***a*, **76***b*, **76***c* and end point **80***a*, **80***b*, **80***c* with respect to time line **66** using keystroke combinations, mouse manipulation, or by any other suitable means as will occur to those of skill in the art. If a clip is shortened, the duration of the source information in corresponding element **28***a* which is used in the project is reduced. If a clip **68** is shifted to the left, with respect to time line **66**, the source information in corresponding element **28** is used earlier within the project than before.

Unlike clips representing 2D or 1D information, clip **68***b* represents 3D animation information, and can be manipulated via NLE system **10** in an enhanced manner, either as a 2D rendering of the 3D animation, if appropriate, or as 3D information which can be manipulated and subsequently re-rendered, if appropriate. For example, clip **68***b* can be lengthened to increase its duration, as with clips for 2D or 1D information, but this can result in element **28***b* being re-rendered from 3D information to a 2D representation. Specifically, if element **28***b* represents an animated 3D model of a bipedal character, such as "John", walking from left to right, lengthening clip **68***b* can result in a decrease of the speed at which the character walks, or an increase in the distance travelled, depending on animation parameters included with the animation information in element **28***b*.

The term "animation parameters", as used herein, is intended to comprise the various components which define an animation element. Animation parameters can be either fixed, i.e.—not being directly changeable via the NLE, or modifiable, i.e.—able to be modified via the NLE. For example, FIG. **5** depicts an animation element **88** for an animated figure walking in a left to right direction. Animation element **88** includes a skeleton parameter **92** comprising the definition of an articulated chain hierarchy and a flexible envelope, or skin, parameter **96** which is applied to skeleton parameter **92**. Both the skeleton parameter **92** and flexible envelope parameter **96** are fixed.

When fixed parameters require modification by an animator, the NLE does allow the animator to invoke the required tool **32**, as specified by the associations for element **88**, as needed. For example, if an animator wishes to modify the articulated chain hierarchy of skeleton parameter **92**, a modelling tool **32** can be invoked from the NLE time line, via a suitable user input event such as double-clicking on the clip, and the skeleton modified as desired. In such a case, Function area **48** can be updated to display at least a subset of the commonly used functions and operators of the modelling tool **32**.

A fixed repeated walk cycle parameter **100** can be defined for a repeatable sequence of skeletal articulations necessary to produce a continuous walking motion for the animated figure represented by element **88**. In addition, a walking speed parameter **104**, a start position parameter **108** and an end position parameter **112** can be defined and set as modifiable parameters associated with the fixed parameters. In some cases, animation parameters are inherently set as being of fixed or modifiable type. For example, the skeleton, flexible envelope and similar parameters can be inherently set as being fixed types as these parameters will not be modified by a director/user in the normal course of creating an edit. Similarly, parameters relating to speed, position and other like factors can inherently be set as being of modifiable types as these parameters will commonly be modified by a director/user in the normal course of creating an edit. In any event, each parameter will have at least one appropriate tool **32** associated with it to permit modification of the parameter. Further, each parameter can explicitly be set by a user as being fixed or modifiable, if such should be desired, by any appropriate user input.

In the NLE of the present invention, element **88** is represented as a clip, such as clip **68***b*, which can be modified in a variety of manners. For example, if the clip for element **88** is lengthened this indicates that the duration of the animation is to be increased. However, the animation corresponding to element **88** can be lengthened in at least two manners. For example, speed parameter **104** can be decreased and start position parameter **108** and end position parameter **112** held constant, with the result that the character in the clip walks the distance between the start and end position with a slower gait, i.e.—the animated character covers the same distance within the increased duration of the clip. Alternatively, speed parameter **104** can be held constant and the position of either start position **108** or end position **112**, corresponding to which end of the clip is altered, is moved to increase the distance travelled by the animated character within the increased duration of the clip.

If the duration of a clip is shortened, corresponding results can be obtained whereby speed parameter **104** is increased, so that the animated character walks the same distance between the start and end positions within the shortened duration, or the start or end positions can be modified so that the animated character walks less distance in the shortened duration.

It is further contemplated that animation elements **28** can have default conditions defined by the animator for modification of duration or for other modifications. For example, in the above-mentioned example of a character walking between a start and end position, the default condition can be to extend beyond the start and end positions, i.e.—to start walking from a position before the start position and/or to walk toward a position beyond the end position. Other appropriate default conditions can be defined for animations, as will occur to those of skill in the art. In this manner, duration or other modifications can result in modifications to the animation in a manner defined by the animator as being appropriate for the particular animation. It is contemplated that the definition of default conditions can be performed, in any suitable manner including the completion of a default condition dialog box by an animator as an explicit step when the animation information is defined, or when the animation information is first used in system **10**.

The actual techniques employed to indicate which method of increasing or decreasing the duration of element **88** is desired are not particularly limited. For example, selecting the clip representing element **88** with a left mouse button can correspond to maintaining the start and end position fixed and changing the speed while selecting the same clip with the right mouse button can correspond to maintaining the speed constant and modifying the start or end position. Other suitable methods for specifying the mapping of changes to a clip to parameters in the corresponding element will be apparent to those of skill in the art.

A modified animation clip may have to be re-rendered to provide a sequence of animation frames in accordance with the desired modification to its animation parameters. In the examples above, where the duration of a clip is increased or decreased, the number and/or content of the frames in the modified element will change and must be re-rendered unless the modification was limited to the discarding of a portion of the animation or the employing of additional frames, previously rendered and stored. Re-rendering of animations in response to modifications can be performed in real time or can be deferred until an appropriate selected time, for example after multiple modifications have been performed and it is desired to view the results. Re-rendering can be performed by one or more rendering engine tools **32** associated with the element requiring re-rendering.

As mentioned above, in addition to modifying the length or position of a clip, the fixed animation parameters of an element can be modified by opening the element, such as by double-clicking on it with a mouse, to invoke an associated tool **32** to access and/or modify the fixed parameters. Further, new fixed or modifiable parameters can be added to a clip by, for example, dragging and dropping functions from function area **48** onto a clip. For example, various effect functions can be provided to modify the result of the animation. A JITTER function can be applied to an animation to add random noise to the motions therein to "jiggle" the rendered animation or a CHARCOAL function can be applied to the animation to give the rendered animation the appearance of having been drawn as a charcoal sketch. Effects or functions can be applied to any clip or to an entire project and can thus affect part of the final edit produced with the present invention or the entire final edit.

In addition to modifying the parameters of any given element in a project, a user may also modify the relationship between elements in a project, and add, subtract or substitute elements within a project. In particular, an edit can be performed with low quality elements, for speed and performance considerations, or elements which are merely place-

holders for information which is not yet available. Once the information becomes available or a final edit is required at a different quality level, the various elements in the project can be replaced by the desired elements without requiring any other effort on the part of the animator. In such a case, the user can select one or more desired elements from a list of available elements presented in the browser in Function area **48** and drag and drop the desired element on top of the clip representing the placeholder or different quality element in a track **72** in NLE time line area **52**. When a desired clip is dropped onto a clip already in a track **72**, the desired clip replaces the clip already in the track, and the start time, end time and duration of the desired clip are set to those of the clip previously in place. Also, any effects or filters which were applied to the clip previously in place are automatically applied to the desired clip.

Further, referring to FIGS. **3** and **4**, in a present embodiment of the invention the relationship, or layering, of elements in a project is determined by the order in which tracks **72** are displayed within NLE **40**. The elements are composited together in the order in which they are displayed from top to bottom, the uppermost element thus forming a first composited layer with successive tracks being composited in respective order as successive layers. Thus, re-ordering the tracks **72** will result in a comparable change to the order in which elements are composited together. It is contemplated that any other suitable means of ordering the layering of elements, such as explicitly assigning a layer to each element, can also be employed if desired and, in such cases, the ordering of tracks **72** in NLE time line area **52** is independent of the compositing order.

Subtracting an element from a project is simply accomplished by removing it from its track and optionally, if it is the only element on its track, removing its track. Adding an element involves dropping an element from a list of available elements or from another track onto an existing track and/or adding a new track to receive the element, as appropriate. Similarly, substituting an element can be accomplished by dropping a new element onto an existing element in a track, as described above.

A special instance of an element in the present invention is a camera. Essentially, a camera is a viewpoint from which a scene is to be rendered. Multiple cameras can be defined for a 3D animation, to allow a director to cut between different viewpoints, and cameras can have positions which change, i.e.—the camera follows a defined path, or is constrained to move with another object, i.e.—the camera can be mounted on the shoulder of an animated character. Camera elements are represented as clips in tracks, like other elements in the NLE system of the present invention, but are principally manipulated by selecting between the available cameras. In some instances, a camera can follow an animated path and manipulation of the clip representing such a camera element can be used to alter the camera's speed along the path. Like other elements in the present invention, if the director should wish to change a camera element in a manner requiring the use of an animation tool, such as changing the path followed by a camera or changing the object to which it is constrained, the associated tool can be invoked from within the NLE system.

The present inventors believe that one of the particular advantages of the present invention is the use of an NLE to manipulate 3D information from a 2D representation thereof. For example, if a 3D animation of a rock music concert is to be produced, the animation artists can have created an element representing the stage (i.e.—a static 3D object); elements representing each of the animated 3D musicians (i.e.—three guitarists and a drummer); an element representing a background (i.e.—a still image of a sky with clouds in it); and an element representing an animated 3D blimp flying over the stage.

In the NLE system in accordance with the present invention, each of these elements can have a track associated with it and the elements are displayed as clips relative to the time line for the edit. While the static elements of the animation, such as the stage, are not manipulated to any great extent within the NLE system, the animated elements can be manipulated as desired. For example, the animators can have defined the 3D animations of the musicians such that the drummer walks onto the stage from the left side and sits down at his drunks and begins to drum. A first guitarist runs onto the stage from the right side, picks up his guitar and starts to play. A second guitarist leaps onto the stage from the left side, runs to his guitar, picks it up and begins to play. The third guitarist runs onto the stage from the right side, stops at the center of the stage and bows to the audience and then turns to pick up his guitar and begins to play. Each musician is defined as an animated bipedal model by the animation artists.

In this example, using the NLE system in accordance with the present invention, the director can change when the various events occur, i.e.—how long after the drummer starts to play that the first guitarist walks onto the stage, by positioning or repositioning the start of the respective clip elements at different points on the time line. The director can change the duration of events, i.e.—how long it requires the drummer to walk across the stage to his drums or how long the third guitarist bows to the audience, by lengthening or shortening the clips to speed up or slow down the animated movement. The director can also re-arrange the compositing of the musician models on the stage, moving the drummer from the back of the stage to the front of the stage, etc. by re-ordering the tracks in the NLE system. The director can also select various cameras from which to view the animation, i.e.—the viewpoint can switch between a camera which is constrained to the moving blimp object, and which provides an aerial view of the stage, and a camera showing a front view of the stage. The director can also turn light sources on or off or otherwise modify them.

Unlike prior art NLE systems, in the examples above the director can actually manipulate the 3D information, via 2D information displayed in the NLE system, to obtain the desired final 2D rendering of the animation. As will be apparent to those of skill in the art, some manipulations of the 2D information displayed in the NLE system will require re-rendering of portions of the animation. For example, slowing the drummer's walk to his drum set will require re-rendering that portion of the animation of the drummer to add the additional frames required. However, as will also be apparent, this will not require re-rendering of unaffected portions of the total animation, i.e.—changing the speed at which the drummer walks does not require re-rendering of the guitarists, although it may require repositioning their clips with respect to the time line. As will be apparent to those of skill in the art, selecting a duration of an element which is less than the entire duration already rendered (i.e.—selecting only five seconds of an eight second animation) will not require any re-rendering, nor will subsequently re-selecting some of the unused rendered portion. Other manipulations can require re-rendering, to a different viewpoint for example, or re-compositing of the rendered elements of the animation.

When re-rendering is required, it is contemplated that a variety of techniques are available to deal with this in a

user-friendly manner. For example, a limited re-rendering can first be performed to produce updated thumbnails on the clips in the NLE system, to provide quick feedback to the director, and a subsequent complete re-rendering of the changes can be performed in the background, either automatically or when explicitly initiated. It is contemplated that this latter case (full re-rendering only when explicitly initiated) may be desired when the director is experimenting with a final cut and a full rendering is not required until a cut of interest has been produced. Further, wire frame or low quality re-renderings can be performed to provide quick feedback to the director and full re-renderings of the desired quality can be subsequently performed. When the director requires a change which requires an animation tool, such as for example changing the path that the blimp flies over the stage, the appropriate tool will be opened by the NLE system to allow the animator or the director to effect the desired changes.

It will be apparent to those of skill in the art that the present invention offers a director or other user a familiar interface for creating a desired final edit, yet the operations performed by the interface can directly map to 3D animation information, rather than to 2D renderings thereof. It is believed that this provides significant advantages over conventional NLE systems which merely manipulate 2D or 1D information.

FIG. **6** shows a representation of a data structure **200** for a project. As shown, each project data structure **200** includes a project description **204**, in this example "Rock Concert", a pointer **208** to a set of project properties, data **212** representing the number of elements in the project and a pointer **216** to those elements **220***a*, **220***b*, . . . **220***n*.

The project properties contain information relevant to the project as a whole and can include total project duration information **224**. This duration information is represented in seconds, microseconds, as a number of frames, etc., as appropriate. The location **228** of the rendered information (video, animation, audio, etc.) and any other relevant information is also present in these properties.

Each element **220***a*, **220***b*, . . . **220***n* points to an appropriate element data structure **232**. Each data structure **232** includes a type identifier **236** which identifies the type of element, i.e.—audio data, video data, still image, 3D animated object, etc., and an element identifier **240** which uniquely identifies that element, i.e.—"lead guitarist", "drummer", the title of the audio track "jump", etc.

Each element also includes an offset **244** which indicates the start of the element relative to the start of the project. For 3D animated objects, this offset is indicated as a number of frames, as is the duration information **248**, while for audio information it is indicated as a time or as a number of samples, as appropriate. Location information **252** indicates where the relevant information is located, including the model definition and animation parameters for a 3D object, the storage location for audio or video information, etc. Additional information **256**, such as a preferred rendering engine and other associations, can also be maintained as appropriate.

As mentioned above, it is also contemplated that projects can themselves be elements of a project. That is, a first project can be created and edited as above and then inserted as an element in a larger project. In the larger project, the first project will be represented as a clip which can be expanded to show the corresponding parameters in an NLE and modified as described above or which can be manipulated as a single clip within the larger product in a similar manner to any other clip. For example, in the above-mentioned Rock Concert animation example, the third guitarist's walking on to the stage can comprise one element, his bow to the audience a second element and his playing his guitar a third element. These three elements can be combined into a single project which is then incorporated into the Rock Concert project. If it is desired to modify or otherwise manipulate the individual elements of the third guitarist's actions, the third guitarist project is opened to access the tracks of the individual elements. Once the desired modifications have been effected, the tracks of the third guitarist's actions can be collapsed again to a project which is represented as a single clip in the overall project of the Rock Concert. This single clip can also be operated on to alter the start time of the third guitarist project (i.e.—to change when he walks onto the stage), or otherwise manipulated as desired.

Though this discussion has focused on modifying 3D animation information, it will be appreciated that the present invention can apply to any other type of information which can be defined as an element having parameters. As each element in a project is independent and can be modified and, if necessary, re-rendered independently of other elements in the project, a variety of elements from different information sources can be edited simultaneously within a single consistent user interface. It will also be apparent that the present invention can be employed in combination with 2D or 1D information as desired. For example, an audio soundtrack can be edited for the rock concert example given above and/or video (such as actual filmed crowd scenes) can be included in the compositing of the final cut.

It will be apparent to those skilled in the art that the foregoing is by way of example only. Modifications, variations and alterations may be made to the described embodiments without departing from the scope of the invention which is defined solely in the claims.

We claim:

1. A method for creating and modifying an edit comprising 3D animation information, comprising:

providing a plurality of animation elements with associated animation parameters, wherein the animation parameters include the 3D animation information;

selecting an animation element from the plurality of animation elements;

generating a clip object by arranging the selected animation element in an associated track for graphical display relative to a timeline;

compositing a plurality of the clip objects so as to form a collection of layered tracks defining the edit, wherein one of the clip objects includes a sequence of 2D animation frames associated with the 3D animation information and having a start position and an end position relative to the time line;

modifying at least one of the animation parameters associated with the selected animation elements and independently re-rendering the associated animation elements in accordance with an extent of each modification, wherein modifying includes modifying a duration of one of the clip objects by altering one of the associated start and end positions of the clip object, and modifying the 3D animation information according to the modified duration of the clip object.

2. The method of claim **1**, wherein said step of modifying said at least one animation parameter comprises the steps of modifying at least one of a first and a last one of the animation frames in said sequence, and re-rendering inter-

vening ones of the animation frames in accordance with an extent of the frame modification.

3. A system for creating and modifying an edit comprising 3D animation information, comprising:

a storage device for storing a plurality of animation elements with associated modifiable animation parameters, wherein the animation parameters include the 3D animation information;

a computer operatively connected to the storage device to access the animation elements stored therein for selecting an animation element from the plurality of animation elements;

means for arranging the selected animation element in an associated track so as to form a clip object for graphical display relative to a time line;

means for compositing a plurality of the clip objects so as to form a collection of layered tracks defining the edit, wherein one of the clip objects includes a sequence of 2D animation frames associated with the 3D animation information and having a start position and an end position relative to the time line;

a tool associated with each animation element for modifying at least one of the animation parameters to initiate independent re-rendering of the animation elements associated with the modified animation parameters in accordance with an extent of each modification, wherein the tool includes means for modifying a duration of one of the clip objects by altering one of the associated start and end positions of the clip object, and means for modifying the 3D animation information according to the modified duration of the clip object.

4. The method of claim 1, wherein each said animation parameters are independently modifiable.

5. The system of claim 1, wherein said step of modifying 3D animation information includes re-arranging said clip in said layered tracks.

6. The system of claim 1, wherein said animation element is a variable angle camera.

7. The method of claim 1, wherein said animation element is the number of light sources.

8. The method of claim 1, wherein said animation information comprises one of static animation information and modifiable animation information.

9. The system of claim 3, further comprising editing data structure for maintaining and co-ordinating the relationship between said animation element and said associated tool.

10. The system of claim 3, wherein said associated tool includes one of an animation program, a composite system, and a rendering machine.

11. The system of claim 3, wherein said animation parameters are independently modifiable.

12. The system of claim 3, wherein said edit is an independent clip of a complex animation.

13. A method for implementing a graphical user interface for editing a three-dimensional animation comprising an animation element with associated animation parameters, wherein the animation parameters define animation of a three-dimensional object over time, the method comprising:

displaying a clip object corresponding to the animation element, wherein the clip object is displayed in a track in a timeline and has a start position and an end position relative to said timeline, wherein the start position and the end position define a duration of the clip object, wherein the duration of the clip object defines the animation parameters associated with the animation element;

rendering the animation element to produce a first rendered sequence of two dimensional animation frames in accordance with the animation parameters as defined by the duration of the clip object;

compositing the first rendered sequence of two-dimensional animation frames with at least one other clip object representing another sequence of two-dimensional frames to produce a representation of the three-dimensional animation;

allowing a user to modify the duration of the clip object corresponding to the animation element by manipulating at least one of the start position and the end position of the clip object, whereby the animation parameters associated with the animation element corresponding to the clip object are modified according to the modified duration of the clip object;

rendering the animation element corresponding to the clip object in accordance with the animation parameters as defined by any modification to the duration of the clip object to produce a second rendered sequence of two dimensional animation frames; and

compositing the second rendered sequence of two-dimensional animation frames with the at least one other clip object representing the other sequence of two-dimensional frames to produce a representation of the three-dimensional animation.

14. The method of claim 13, wherein the three dimensional object is defined by information defining a skeleton and skin of the three dimensional object in three dimensions and the animation is defined by position and motion information of the three dimensional object in three dimensions.

15. The method of claim 13, further comprising modifying the animation parameters according to the modified duration of the clip object such that speed of the three dimensional object in the animation is modified.

16. The method of claim 13, further comprising modifying the animation parameters according to the modified duration of the clip object such that a position of the three dimensional object in the animation is modified.

17. The method of claim 1, wherein the animation parameters include information defining a skeleton and skin of a three dimensional object in three dimensions and position information of the three dimensional object in three dimensions over time.

18. The method of claim 1, wherein modifying the 3D animation information comprises modifying the animation parameters according to the modified duration of the clip object such that speed of motion of the three dimensional object is modified.

19. The method of claim 1, wherein modifying the 3D animation information comprises modifying the animation parameters according to the modified duration of the clip object such that a position of the three dimensional object is modified.

20. The system of claim 3, wherein the animation parameters include information defining a skeleton and skin of a three dimensional object in three dimensions and position information of the three dimensional object in three dimensions over time.

21. The system of claim 3, wherein the means for modifying the 3D animation information comprises means for modifying the animation parameters according to the modified duration of the clip object such that speed of motion of the three dimensional object is modified.

22. The system of claim 3, wherein the means for modifying the 3D animation information comprises means for modifying the animation parameters according to the modified duration of the clip object such that a position of the three dimensional object is modified.

* * * * *